YOUR
COMMODORE
C16 and Plus/4
The Essential Guide

FREE with

YOUR
COMMODORE
NOVEMBER 1988

Games Reviewed:
Bombjack
Thrust
Arthur Noid
Omnibus
Storm

UNBEATABLE PROGRAMS:

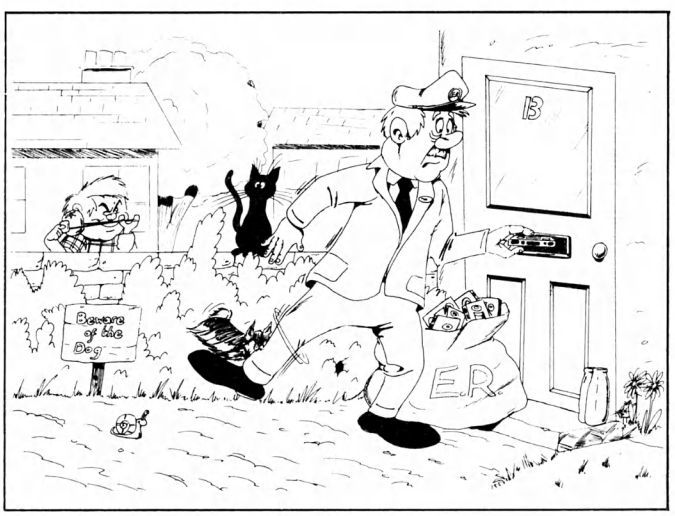
MONEY+ △ TEXT 80 △ PLUS/4 CONVERTER △ DATA FILE

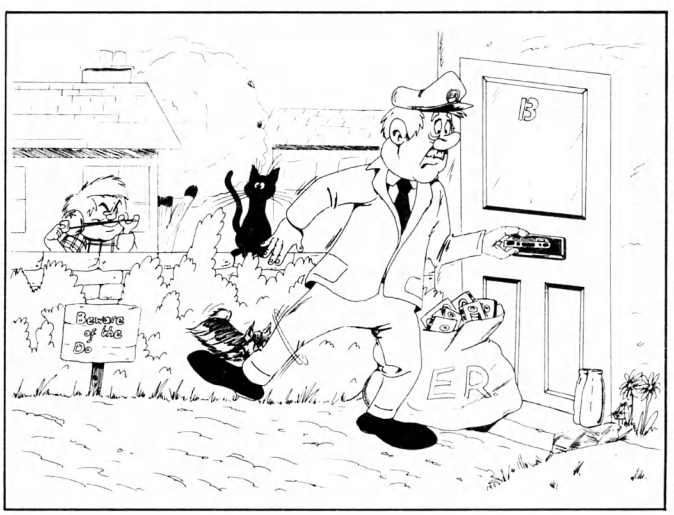
Win a bundle of software

Your chance to win the games reviewed in this supplement

To celebrate this special C16/Plus/4 supplement we've gathered together all the games we could find. We reviewed some of them inside and included them all in our game finder. Now we want to give them all away.

One lucky reader will receive a bundle of over 50 games including Frank Bruno's Boxing, Arthur Noid, Saboteur and Auf Wiedersehen Monty. All you have to do is spot the difference.





Plus/4 Entry coupon

Name .....

Address .....

..... Postcode .....

Number of differences found .....

Closing date: November 30th

Post to: Your Commodore Plus /4 Competition, 1 Golden Square, London W1R 3AB

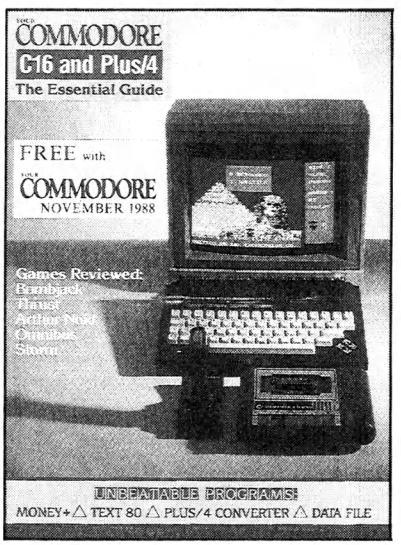
Contents

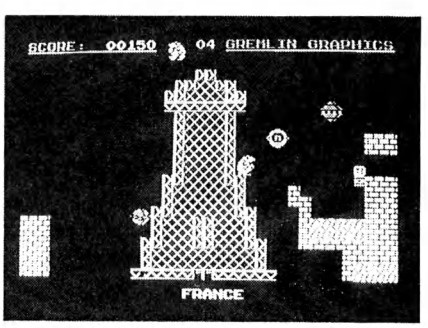
ARGUS PRESS GROUP

Editor: Stuart Cooke
Deputy Editor: Tony Hetherington
Technical Editor: Eric Doyle
Origination: Ebony Typesetting
Artist: Alan Batchelor
Designer: Neil Sweetman

Argus Specialist Publications Limited Editorial & Advertisement Office, Your Commodore, No 1 Golden Square, London W1R 3AB. Telephone: 01-437 0626 Telex: 8811896.

FEATURES





PROGRAMS

Money/Plus

Give your PLUS/4 powerful budget organizing facilities.

Have you ever wanted to use a computer to organize your financial affairs but found the software or firmware too complex to manage? Did it take longer using the computer than the back of an envelope? Did your bank manager refuse a loan for a new Amiga because you had a disorganized bugeting system? Perhaps *MONEY/PLUS* will solve some or all of your financial problems. I have used this program for the last three years and it has transformed budgeting into a quick and easy task. Once a list of monthly income sources and expenses has been made and the appropriate data entered, your PLUS/4 will do the rest.

Getting Started.

Type in the program exactly as listed. There are no print statements containing substitutes for cursor control or colour changes etc. as these have all been programmed with CHR$(number) codes. Some program lines include text like " UP " or " LEFT " and should be typed in exactly as shown and not as cursor up or cursor left print control characters. Save a copy of the program for future use. Lines 1120 to 1230 hold the names of expenses and income sources. LIST the line that you may wish to change and enter your own names. E$(01) is the first string variable and contains the name of your first expense. E$(02) contains the name of your second expense, and so on. The names that you choose should be no more than twelve characters long, but should also be made up to twelve characters in length using the appropriate number of spaces in between the inverted commas. The names of income sources are held in the string variable I$(01),I$(02) etc. Don't forget to press the RETURN key to enter each changed program line into your program. Save a copy of your new program as your working copy under a suitable name like "MONEY/PLUS USER". If thirty expenses and five sources of income do not meet your needs, you will have to change the values of I and E in line 1040 to your own needs. E is your number of expenses PLUS one. So, for fifty expense items, change E to 51. To increase your number of income sources, change I on line 1040 to the exact number of incomes that you might want to use. Lines like 1210 and 1230 will need to be added with any appropriate twelve character names, or blank names, as mentioned above. You will need to save a copy of your amended program under a suitable name. Once you have made your personal changes, the program is ready to RUN.

Using The Program

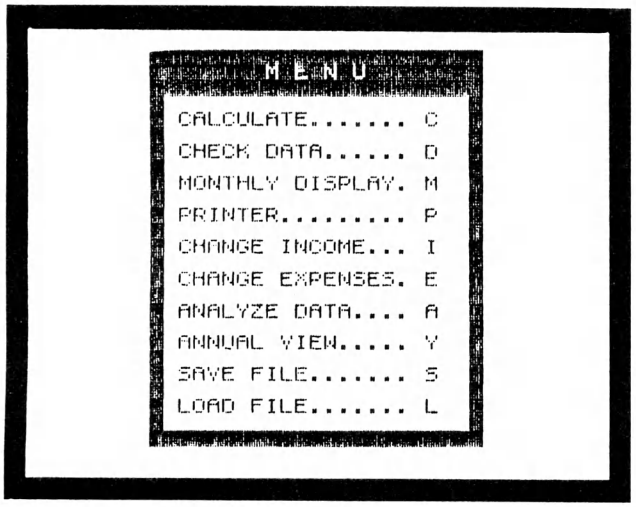
Load your program and RUN it. A MENU screen, see fig. 1, appears. To enter your arithmetic data for expenses or income, press key E or I. Screen 2 will appear ready to enter your data. The four cursor keys are used to select the months and items of expense or income to be checked or changed, as required. To select the INCOME data entry screen press Key I and press key E to select the EXPENSES data entry screen. When the month and expense or income have been selected, the value of that item is show. To change the value of this selected item, press key C. A flashing cursor appears for the new value to be entered. The value entered is to the nearest pound, and four digit numbers up to £9999 can be entered. If you try to enter £12345, only £1234 will be entered. To enter a number, type it in and press the RETURN key. If you have an item of expense or income that is to be the same for all twelve months, type in the number and press key A and not the RETURN key. After each item of data is entered, use the cursor keys to find the next item to ckeck or change. Your data will be formatted for printing as shown in fig. 2. Each item has its own row number and this is shown, with the name of the item, in the data entry screen. This will enable you to quickly scan through the items and months after you have produced your first printed copy of your annual buget. To quit data entry or checking mode, press key M. If you have entered data, or if the first month of your budget has not been set in the calculation routine, the program will enter the caculation routine. The pressing of key M will otherwise return the program to the MENU screen.

Calculation Routine

When data entry is completed from either the data entry screen or a tape or disk file, the new data will be processed by the routine in lines 2000 to 2150. The routine will ask for and set the first month of your buget. Entering month number six will cause the buget to go from June through to May. Once set, you need only press the RETURN key when asked for the first month unless, of course, you wish to change it. After processing the data, screen 3 will appear to show your total monthly credits, debits and balances and their annual totals.

Data presentation

Screen 3, shown after each calculation routine, or by pressing key Y in screen 1 or screen 4, displays the monthly total values for a twelve month period, starting with the first month as selected. To view an individual month's details, screen 4 is entered from the MENU screen with the pressing of key M. Screen 4 displays all those items of expenses and income



SCREEN 1

Fig. 1 SCREEN DISPLAYS.

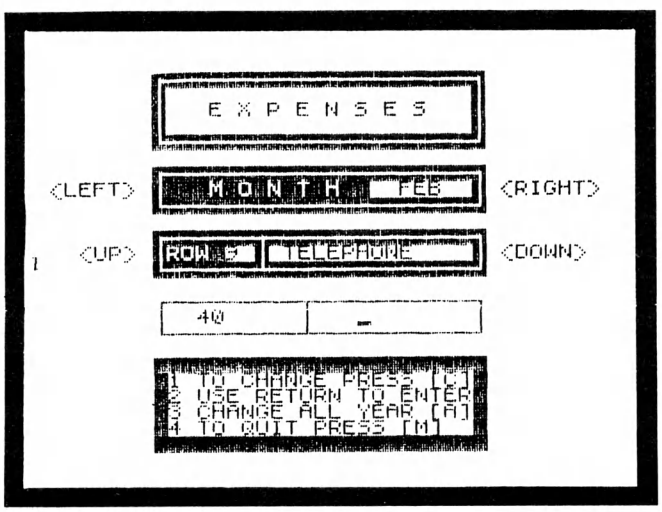
1 Menu.

2 Data entry for expenses and income.

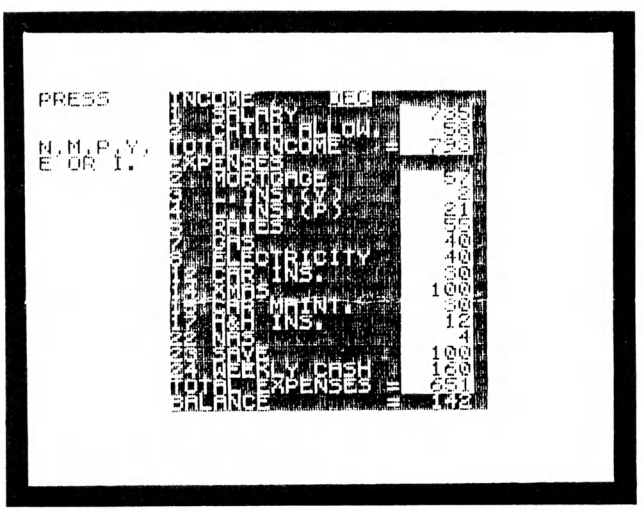
3 Annual statement of the monthly totals starting at December.

4 Monthly preview showing income, expenses and balance.

5 Analysis of expenses or income, showing annual total.



SCREEN 2



SCREEN 4

SCREEN 3

DEC	793	651	142
JAN	807	668	139
FEB	793	638	155
MAR	793	623	170
APR	793	648	145
MAY	807	668	139
JUN	808	653	155
JUL	808	676	132
AUG	822	673	149
SEP	808	653	155
OCT	833	643	190
NOV	847	683	164
	9712	7877	1835

(PRESS M,P,E OR I)

SCREEN 5

ANALYSIS OF

RATES 5

JUL	70
AUG	70
SEP	70
OCT	70
NOV	70
DEC	70
JAN	70
FEB	0
MAR	0
APR	77
MAY	77
JUN	77
TOTAL	721

CURSOR <UP> OR <DOWN>

ANALYSIS OR MENU? (A/M)

that have data entry. Any zero value item is omitted. The total income, expense and a balance are shown. Here is the limiting factor on the number of monthly items that can be entered and displayed, without the screen scrolling upwards. If this is a problem, using the COMMODORE key slows down printing and screen rolling. The month shown is the first month of your buget. To see the following months, press key N. You can exit from screen 4 to the MENU by pressing key M or you can go direct to screen 3 with key Y, screen 2 with key E or I, or you can go direct to the printing mode with key P. these keys are shown on the screen, as are the key prompts for the other screens. The last screen to describe is a facility for analizing individual expense or income items. Pressing key A in MENU mode will give you the choice of which items to analize. If you chose to look at expense item number 5, your monthly commitments are displayed, starting at the first month of your budget, and an 'annual total for item 5. To view other expenses, you can move up and down your list with the cursor up and cursor down keys. To analize income, press key A and choose which item you want. In both expense and income analysis, response to which item of analysis you require of the RETURN key only, will start the analysis at item 1. You exit screen 5 to MENU mode with key M.

Saving And Loading Files.

Having set up a buget, it will be used over and over again with occasional updating as your financial circumstances change. Saving a file of your buget is very important as it will enable your future planning to be done very speedily. Just recall a *tape or disk* file and modify it to your current needs, without having to start from scratch. So, having produced your masterpiece of financial wizardry, save a file for future use. From MENU mode, press key 5 and choose tape or disk for your file. Tape files are recorded over the section of tape that you choose on the cassette, so take care not to erase any wanted files or programs already on the tape. Disk files are always stored with the same file name, so a file will thus automatically overwrite a previous *MONEY/PLUS* file on the disk. This is a simple way of keeping a file without the need for file names to be issued or remembered! If you wish to keep several files, use a separate disk for each one. As you will need to use the exact program that saved a file to recall it, it is wise to keep one disk for each version of your program and its own file. Remember that lines 1120 to 1230 contain a list (file) of the names of your expense and income. Saving and loading of files does not use that list of names. If you wish to keep lots of different files, and easily change the names of each item, the program can be easily changed to save and load a list of names. Lines 11000 and 12110 control loading and saving of files. Using a 1541 disk drive over the last three years, I have had totally trouble free keeping of files. If I want to file an amended buget and keep the original as well, I just use two disks. The data entry routine is so quick and easy to to use that I don't even keep a backup file. For tape users, I would advise a tape just long enough to store a copy of *MONEY/PLUS* on one side, with the record protect tag removed, and your file on the reverse side. This way you keep the file with its program and you can load the program, turn over the cassette, and load its file.

Printing A Budget.

The format chosen for printing is suitable for 80 column dot matrix printers and the 165 column DPS 1101 daisy wheel printer. You can date your buget copies and you can have a full annual statement or individual monthly statement. See fig. 2. If you have a DPS 1101, you can print the two statements alongside each other using page number 1 for one of them, rewind the paper to the top, and use page number two for the other statement. Set the character pitch to 15cpi though, or you will print a mess! The annual and monthly statements can be printed on either page 1 or page two. If you don't like the double width characters on the monthly dot matrix statements, remove the "PRINTCHR$(14);" part of line 7030. The names of items are shortened to eleven characters on annual statements to fit everything onto 80 column dot matrix printers. The program returns to MENU mode after printing a statement. It is the 80 columns on most printers that limits the individual items to a maximum of four digits or £9999. If you have more columns on your printer and the financial situation that requires larger numbers, it would not be too big a job to change the program to your own needs. Once you have produced your own printed statements, it makes any future data input or buget amendment easier to do by reference to printed names, row number and numerical values. Screen presentation and printed format are well matched to make planning and any future amendments easy. If you don't have a printer, *MONEY/PLUS* can still do your financial planning and present the information on the screen. I find "what if I spend so much" type planning easy, quick and even fun to do.

Hints On Use.

Try to keep the the number of items of expense and income to a minimum. This will enable all the monthly items to appear in screen 4, as well as reducing filing, printing and calculating times. Lump together items so that separate items like rent on one line and rates on another line can be combined as rent & rates; still twelve characters for the joint name. Items of financial significance that do not need a separate line can be lumped together on lines called extra expense 1, extra expense 2 etc. Let's hope all your little extras don't come in the same month! When you make your own list of names for expenses and income, put them in the order that you

want so that all related items appear together. Put your housing costs together, services like gas, electricity and telephone next to each other, and so on. This gives you less chance of missing something out and a better idea of where all that money goes! Carefully add your names to lines 1120 to 1230. DO NOT leave any gaps in the names. If you have 25 items of expenditure, write them into E$(01) to E$(25) without any gaps in the order that you want them to appear on your statements. E$(12) is printed above and next to E$(13).

C64 Conversion.

Very little work would be needed to make this program work on a C64 computer. CHR$(number) codes for screen printing controls like colour or flashing characters can be changed to print in your own colour choice and to leave out flash on and off controls. To do this, the codes are listed below.

CHR$(130)	Flash on.
CHR$(132)	Flash off.
CHR$(155)	Light green.
CHR$(153)	Light blue.

Screen background and border is black. C64 does not support IF, THEN, ELSE statements, so where these are used the logic will need extra lines of IF, THEN statements. Experienced BASIC users should have no difficulty.

PROGRAM: MONEY PLUS

```
1000 REM "MONEY/PLUS" TAPE/DISK
BUDGET PROGRAM FOR PLUS/4.  P.G.
SIMONDS, DEC-1987
1010 REM BUDGETS TO NEAREST POUN
D FOR EXPENSES AND INCOMES UP TO
 \9999.
1020 REM ANNUAL TOTALS UP TO \99
999.
1030 REM PRINTOUTS CAN START AT
ANY MONTH OF THE YEAR.
1040 E=31:I=5:LS="":FORP=1TO21:L
$=LS+CHR$(192):NEXTP
1050 DIME(12,E):DIMI(12,E):DIME$
(E):DIMI$(E):DIMMO$(12):DIMET(12
):DIMIT(12)
1060 DIMMT(12):DIMRT(12):SPS="
                              "
1070 N=0:E=E-1:HS=CHR$(176)+LS+C
HR$(174):BS=CHR$(173)+LS+CHR$(25
3)
1080 COLORO,1,0 :COLOR1,2:COLOR4
,1
1090 MO$(1)="JAN":MO$(2)="FEB":M
O$(3)="MAR":MO$(4)="APR":MO$(5)=
"MAY"
1100 MO$(6)="JUN":MO$(12)="DEC"
1110 MO$(7)="JUL":MO$(8)="AUG":M
O$(9)="SEP":MO$(10)="OCT":MO$(11
)="NOV"
1120 E$(01)="LOAN          ":E$(02
)="MORTGAGE     ":E$(03)="L.INS.(
V)   "
1130 E$(04)="L.INS.(P)    ":E$(05
)="RATES        ":E$(06)="WATER
    "
1140 E$(07)="GAS           ":E$(08
)="ELECTRICITY ":E$(09)="TELEPHO
NE    "
1150 E$(10)="CAR TAX       ":E$(11
)="CAR MOT      ":E$(12)="CAR INS
.    "
1160 E$(13)="HOLIDAYS      ":E$(14
)="XMAS         ":E$(15)="CAR MAI
NT.  "
1170 E$(16)="CARAVAN CLUB":E$(17
)="H&H INS.     ":E$(18)="TV LICE
NCE  "
1180 E$(19)="MAYDAY        ":E$(20
)="HOUSE MAINT.":E$(21)="CLOTHES
    "
1190 E$(22)="NAS           ":E$(23
)="SAVE         ":E$(24)="WEEKLY
CASH "
1200 E$(25)="OTHER EXP.    ":E$(26
)="             ":E$(27)="
    "
1210 E$(28)="              ":E$(29
)="             ":E$(30)="
    "
1220 I$(01)="SALARY        ":I$(02
)="CHILD ALLOW.":I$(03)="OTHER I
NC.  "
1230 I$(04)="              ":I$(05
)="             "
1240 GOTO4000
2000 REM CALCULATION ROUTINE
2010 N=1:Z=0:Y=0
2020 FORP=1TO12:ET(P)=0:FORR=1TO
E:ET(P)=ET(P)+E(P,R):NEXTR:Z=Z+E
T(P):NEXTP
2030 FORP=1TO12:IT(P)=0:FORR=1TO
I:IT(P)=IT(P)+I(P,R):NEXTR:Y=Y+I
T(P):NEXTP
2040 FORP=1TO12:MT(P)=IT(P)-ET(P
):NEXTP
2050 PRINTCHR$(147):FORP=1TO6:PR
INT:NEXTP
2060 INPUT"          1ST MONTH NU
MBER";SM:IFSM<1ORSM>12THEN2050
2070 RR=SM
2080 RT(SM)=MT(SM):X=RT(SM)
2090 FORP=1TO11
2100 SM=SM+1:IFSM=13THENSM=1
2110 RT(SM)=X+MT(SM):X=RT(SM)
2120 NEXTP
2130 IFRT(SM)=Y-ZTHEN2150
2140 PRINT"INCORRECT CALCULATION
!":STOP
2150 SM=RR
3000 REM ANNUAL STATEMENT FOR SC
REEN
3010 UL$=CHR$(159)+"    ===== ==
=== =====";PRINTCHR$(147);:PRINT
SP$;
3020 PRINTCHR$(159);"   ";CHR$(1
8);" ANNUAL STATEMENT ";CHR$(145
):PRINT
3030 PRINTSP$;CHR$(155);"    ";CH
R$(18);"CREDIT";CHR$(146);"  ";
3040 PRINTCHR$(18);"DEBIT";CHR$(
146);"  ";CHR$(18);"TOTAL";CHR$(1
45):PRINT
3050 FORP=1TO12:PRINTSP$;CHR$(15
8);CHR$(18);MO$(SM);CHR$(146);CH
R$(5);
3060 PRINTRIGHT$("       "+STR$(I
T(SM)),6);RIGHT$("       "+STR$(E
T(SM)),5);
3070 PRINTRIGHT$("       "+STR$(M
T(SM)),6)
3080 SM=SM+1:IFSM=13THENSM=1
3090 NEXTP
3100 PRINTSP$;UL$;CHR$(5)
3110 PRINTSP$;RIGHT$("          "
+STR$(Y),9);
3120 PRINTRIGHT$("       "+STR$(Z
),6);
3130 PRINTRIGHT$("       "+STR$(Y
-Z),6)
3140 PRINTSP$;UL$:PRINT:PRINT:PR
INTSP$;" (PRESS M,P,E OR I)"
3150 GETA$
3160 IFA$=""THEN3150
3170 IFA$="M"THEN4000
3180 IFA$="P"THEN5000
3190 IFA$="E"THENSM=RR:BB$="E":G
OTO8000
3200 IFA$="I"THENSM=RR:BB$="I":G
OTO8000
3210 GOTO3150
4000 REM MENU
4010 A1$=CHR$(18)+" "+CHR$(146)+
CHR$(5)+" "
4020 A2$=" "+CHR$(153):A3$="  "+
CHR$(18)+CHR$(155)+" "
4030 A3$="  "+CHR$(18)+CHR$(155)
+" "
4040 A$=CHR$(18)+" "+CHR$(146)+"
                  "+CHR$(18)+
" "
4050 PRINTCHR$(147);CHR$(155);SP
$;CHR$(18);"
    "
4060 PRINTSP$;CHR$(155);CHR$(18)
;"        M E N U         "
4070 PRINTSP$;CHR$(18);"
                  ";:PRINTSP$;A$
4080 PRINTSP$;A1$;"CALCULATE....
...";A2$;"C";A3$:PRINTSP$;A$
4090 PRINTSP$;A1$;"CHECK DATA...
...";A2$;"D";A3$:PRINTSP$;A$
4100 PRINTSP$;A1$;"MONTHLY DISPL
AY.";A2$;"M";A3$:PRINTSP$;A$
4110 PRINTSP$;A1$;"PRINTER......
...";A2$;"P";A3$:PRINTSP$;A$
4120 PRINTSP$;A1$;"CHANGE INCOME
...";A2$;"I";A3$:PRINTSP$;A$
4130 PRINTSP$;A1$;"CHANGE EXPENS
ES.";A2$;"E";A3$:PRINTSP$;A$
4140 PRINTSP$;A1$;"ANALYZE DATA.
...";A2$;"A";A3$:PRINTSP$;A$
4150 PRINTSP$;A1$;"ANNUAL VIEW..
...";A2$;"Y";A3$:PRINTSP$;A$
4160 PRINTSP$;A1$;"SAVE FILE....
...";A2$;"S";A3$:PRINTSP$;A$
4170 PRINTSP$;A1$;"LOAD FILE....
...";A2$;"L";A3$:PRINTSP$;A$
4180 PRINTSP$;CHR$(18);"
                  "CHR$(146);CHR$(1
9)
4190 GETBB$
4200 XX=0
4210 IFBB$="C"ANDKK=1THENPRINTCH
R$(147);CHR$(5);:GOTO4310
4220 IFBB$="I"ORBB$="E"THEN8000
4230 IFBB$="D"THENBB$="E":GOTO80
00
4240 IFBB$="S"ANDKK=1THEN11000
4250 IFBB$="M"ANDN=1THEN9000
4260 IFBB$="Y"ANDN=1THEN3000
4270 IFBB$="P"ANDN=1THEN5000
4280 IFBB$="A"ANDN=1THEN10000
4290 IFBB$="L"THEN12000
4300 GOTO4190
4310 FORP=1TO7:PRINT:NEXTP:PRINT
"              ";CHR$(18);CHR$(13
0);
4320 PRINT" CALCULATING. ";CHR$(
132);CHR$(146):GOTO2000
5000 REM PRINTER ROUTINE
5010 P1$="                    "
:P2$=P1$+P1$+P1$+P1$:PRINTCHR$(1
47):PRINT
5020 PRINTCHR$(155);"    IS PRIN
TER CONNECTED? (Y/N)":GETKEYY$:I
FY$<>"Y"THEN4000
5030 PRINT:PRINT:PRINT"    IS PR
INTER LOADED? (Y/N)":GETKEYY$:IF
Y$<>"Y"THEN4000
5040 PRINT:PRINT:PRINT"    IS PR
INTER TURNED ON? (Y/N)"
5050 GETKEYY$:IFY$<>"Y"THEN4000
5060 PRINTCHR$(5):PRINT:INPUT"
  DATE";DT$
5070 PA=1:PRINT:PRINT:INPUT"
PAGE NUMBER";PA
5080 PRINT:PRINT:PRINTCHR$(155);
"    ANNUAL/MONTHLY STATEMENT? (
A/M)":GETKEYCC$
5090 IFCC$="A"THEN6000
5100 IFCC$="M"THEN7000
5110 GOTO4000
6000 REM PRINT ANNUAL BUDGET
6010 OPEN4,4:CMD4:GOSUB6360
6020 PRINT"ANNUAL BUDGET ";DT$:S
=LEN(DT$)+14:GOSUB6360
6030 FORP=1TOS:PRINT"-";:NEXTP:P
RINT
6040 LU$="-------------- ---- --
-- ---- ---- ---- ---- ---- ----
 ---- ----"
6050 LU$=LU$+" ---- ---- -----":
GOSUB6360
6060 PRINT"INCOME          ";:GOSU
B6390:GOSUB6360:PRINTLU$:SM=RR:F
ORP=1TO1
6070 IFI$(P)="              "THEN6
120
6080 GOSUB6360:PRINTMID$(STR$(P)
+"  ",2,3);LEFT$(I$(P),11);:SS=0
:FORM=1TO12
6090 D=I(SM,P):SS=SS+D:PRINTRIGH
T$("    "+STR$(D),5);
6100 SM=SM+1:IFSM=13THENSM=1
6110 NEXTM:D=SS:PRINTRIGHT$("
   "+STR$(D),6)
6120 NEXTP:GOSUB6360:PPINTLU$
6130 GOSUB6360:PRINT"TOTALS
   ";:SS=0:FORM=1TO12:D=IT(SM):S
S=SS+D
6140 PRINTRIGHT$("    "+STR$(D),
5);:SM=SM+1:IFSM=13THENSM=1
6150 NEXTM:D=SS:PRINTRIGHT$("
   "+STR$(D),6):GOSUB6360:PRINTL
U$:PRINT:PRINT
6160 PRINT:GOSUB6360:PRINT"EXPEN
SES        ";:GOSUB6390:GOSUB6360:
PRINTLU$
6170 SM=RR:FORP=1TOE:IFE$(P)="
          "THENS220
6180 GOSUB6360:PRINTMID$(STR$(P)
+"  ",2,3);:PRINTLEFT$(E$(P),11)
;:SS=0
6190 FORM=1TO12:D=E(SM,P):SS=SS+
D:PRINTRIGHT$("    "+STR$(D),5);
6200 SM=SM+1:IFSM=13THENSM=1
6210 NEXTM:D=SS:PRINTRIGHT$("
   "+STR$(D),6)
6220 NEXTP
6230 GOSUB6360:PRINTLU$:GOSUB636
0:PRINT"TOTALS          ";:SS=0
6240 FORM=1TO12:D=ET(SM):SS=SS+D
:PRINTRIGHT$("    "+STR$(D),5);
6250 SM=SM+1:IFSM=13THENSM=1
6260 NEXTM:D=SS:PRINTRIGHT$("
   "+STR$(D),5):GOSUB6360:PRINTL
U$:PRINT:PRINT
6270 PRINT:GOSUB6360:PRINT"
          ";:GOSUB6390:GOSUB6360:
PRINTLU$
6280 GOSUB6360:PRINT"BALANCES
   ";:SS=0:SM=RR:FORM=1TO12
6290 D=MT(SM):SS=SS+D:PRINTRIGHT
$("    "+STR$(D),5);:SM=SM+1:IFS
M=13THENSM=1
6300 NEXTM:D=SS:PRINTRIGHT$("
   "+STR$(D),6):GOSUB6360:PPINTL
U$:PRINT:PRINT
6310 PRINT:GOSUB6360:PRINT"
          ";:GOSUB6390:GOSUB6360:
PRINTLU$
6320 GOSUB6360:PRINT"ACCUMULATIO
N   ";:SM=RR:FORM=1TO12:D=PT(SM)
6330 PRINTRIGHT$("    "+STR$(D),
5);:SM=SM+1:IFSM=13THENSM=1
```

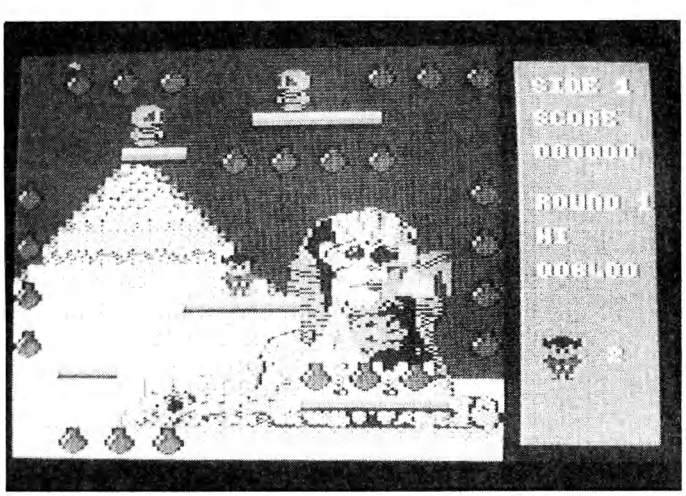
```
6340 NEXTM:D=Y-Z:PRINTRIGHT$("
     "+STR$(D),6):GOSUB6360:PRINT
LU$
6350 PRINT#1:CLOSE1,4:GOTO1000
6360 REM PRINT BLANK PAGE ONE SU
BROUTINE
6370 IFPA=2THENPRINTP2$;
6380 RETURN
6390 REM PRINT MONTHS SUBPOUTINE
6400 SM=PR:FORP=1TO12:PRINT"  ";
MO$(SM);:SM=SM+1:IFSM=13THENSM=1
6410 NEXTP:PRINT" TOTAL":RETURN
7000 REM PRINT MONTHLY STATEMENT
7010 UU$="----------------------
";PRINTCHR$(5):PRINT
7020 INPUT"    NUMBEP OF MONTH T
O BE PRINTED";M:IFM<1ORM>12THEN4
000
7030 OPEN4,4:CMD4:PRINTCHR$(14);
:GOSUB6360:PRINTDIS:S=LEN(DI$):G
OSUB6360
7040 FORP=1TOS:PRINT"=";:NEXTP:P
RINT:PRINT:GOSUB6360
7050 PRINT"ESTIMATES FOR THE MON
TH OF ";MO$(M):GOSUB6360
7060 PRINT"----------------------
----------":PRINT:GOSUB6360
7070 PRINT"INCOME:-
\":GOSUB6360:PRINIUU$:FORP=1TOI
7080 IFI(M,P)=OOR1$(P)=""THEN710
0
7090 GOSUB6360:PRINII$(P);"
  ";RIGHI$("     "+STP$(II(M,P)),4)
7100 NEXTP:GOSUB6360:PRINIUU$:GO
SUB6360
7110 PRINT"IOIAL INCOME     ";PIG
HI$("     "+STP$(II(M)),6):GOSUB6
360
7120 PRINIUU$:PRINT:GOSUB6360:PR
INT"EXPENSES:-      \"
7130 GOSUB6360:PRINTUU$:FORP=1TO
E
7140 IFE(M,P)=OOPE$(P)=""THEN716
0
7150 GOSUB6360:PRINIE$(P);"
  ";RIGHI$("     "+STP$(EI(M,P)),4)
7160 NEXTP
7170 GOSUB6360:PRINTUU$:GOSUB636
0
7180 PRINT"TOIAL EXPENSES  ";PIG
HI$("     "+STP$(EI(M)),6):GOSUB6
360
7190 PRINTUU$:PRINT:GOSUB6350:PR
INTUU$:GOSUB6360
7200 PRINT"BALANCE          ";RIG
HI$("     "+STR$(MI(M)),6)
7210 GOSUB6350:PRINTUU$:PRINT:GO
SUB6350:PRINTUU$:GOSUB6360
7220 PRINT"ACCUMULATION     ";RIG
HI$("     "+STP$(PI(M)),6):GOSUB6
350:PRINTUU$
7230 PRINT#4,CHP$(15):PRINT#4:CL
OSE4,4:GOTO4000
8000 REM DATA CHECK AND CHANGE R
OUTINE
8010 FD=0:IFBB$="I"THENMD$="I":E
LSEMD$="E"
8020 CA$=CHR$(5)+CHR$(18)+CHR$(1
30)+CHR$(157)+CHR$(157)+CHR$(157
)
8030 CA$=CA$+"CALCULATING"+CHR$(
132)+CHR$(146)
8040 R$="":FORP=1TO24:R$=R$+CHR$
(29):NEXTP
8050 CO=SM:KK=1.PO=1:IFSM=OTHENS
M=1:CO=SM
8060 C$=CHR$(158)+CHR$(130)+CHR$
(175)+CHR$(132)+CHR$(5)+CHR$(157
)
8070 IFFD=OTHENGOTO8440
8080 GETKEYHH$:IFHH$=CHR$(145)TH
ENRO=RO-1:IFRO<1THENRO=1:GOTO835
0
8090 IFHH$=CHR$(145)THEN8350
8100 IFHH$=CHR$(17)ANDBB$="E"AND
RO<ETHENRO=RO+1:GOTO8350
8110 IFHH$=CHR$(17)ANDBB$="I"AND
RO<ITHENRO=RO+1:GOTO8350
8120 IFHH$=CHR$(29)THENCO=CO+1:I
FCO>12THENCO=1:GOTO8350
8130 IFHH$=CHR$(157)THENCO=CO-1:
IFCO<1THENCO=12:GOTO8350
8140 IFHH$=CHR$(29)ORHH$=CHR$(15
7)THEN8350
8150 IFHH$="E"ANDBB$="E"THEN8080
8160 IFHH$="E"THENBB$="E":RO=1:C
O=SM:GOTO8350
8170 IFHH$="I"ANDBB$="I"THEN8080
8180 IFHH$="I"THENBB$="I":RO=1:C
O=SM:GOTO8350
8190 IFHH$="C"THENN=0:PRINTC$;:B
M$="":GOTO8240
8200 IFHH$="M"ANDN=1THEN4000
8210 IFHH$="Y"ANDN=1THENSM=PR:GO
TO3000
8220 IFHH$="M"THENPRINTCA$:GOTO2
000
8230 GOTO8080
8240 GETKEYNU$:IFNU$=CHR$(13)THE
N8280:ELSEIFNU$="A"THEN8310
8250 IFASC(NU$)-48<CORASC(NU$)-4
8>10THEN8240
8260 BN$=BN$+NU$:IFLEN(BN$)>4THE
N8280
8270 PRINTNU$;:C$=:GOTO8240
8280 NU=VAL(LEFT$(BN$,4))
8290 IFBB$="E"THENE(CO,RO)=NU:NU
=0:GOTO8350
8300 IFBB$="I"THENI(CO,RO)=NU:NU
=0:GOTO8350
8310 NU=VAL(LEFT$(BN$,4))
8320 FORP=1TO12
8330 IFBB$="E"THENE(P,RO)=NU:ELS
EIFBB$="I"THENI(P,RO)=NU
8340 NEXTP:NU=0
8350 PRINTCHR$(19);CHR$(5);CHR$(
146):PRINT:PRINT:PRINTLEFT$(R$,1
3);
8360 IFBB$="E"THENPRINT"E X P E
N S E S":ELSEPRINT"  I N C O M E
  "
8370 PRINT:PRINT:PRINT:PRINT:PRI
NTP$;CHR$(18);CHR$(158);"  ";MO$
(CO);"  "
8380 PRINT:PRINT:PRINT:PRINTLEFT
$(R$,14);CHR$(153);MID$(STR$(RO
)+"  "),2,2);
8390 IFBB$="E"THENPRINTLEFT$(R$,
2);E$(RO):ELSEPRINTLEFT$(R$,2);I
$(RO)
8400 PRINT:PRINT:PRINT:PRINTLEFT
$(R$,12);CHR$(153);
8410 IFBB$="E"THENPRINTMID$((STR
$(E(CO,RO))+"      "),2,5);CHR$(2
9);
8420 IFBB$="I"THENPRINTMID$((STR
$(I(CO,RO))+"      "),2,5);CHR$(2
9);
8430 PRINTLEFT$(R$,5);"      ";:F
ORP=1TO5:PRINTCHR$(157);:NEXTP:G
OTO8080
8440 FD=1:PRINTCHR$(147);PRINTCH
R$(155);SP$;CHR$(18);H$
8450 ED$=SP$+CHR$(18)+CHR$(125)+
CHR$(146)+"
"+CHR$(18)
8460 ED$=ED$+CHR$(125):PRINTED$:
PRINTED$:PRINTED$
8470 PRINTSP$;CHR$(18);B$:PRINT
8480 PRINTSP$;CHR$(18);H$
8490 PPINT"  ";CHR$(158);"<LEFT>
 ";CHP$(155);CHP$(18);CHR$(125);
8500 PRINT"  M O N T H  ";CHR$(
146);"      ";CHR$(18);CHR$(125
),CHR$(146);
8510 PRINTCHR$(158);" <RIGHT>";C
HR$(155)
8520 PRINTSP$;CHP$(18);E$:PRINT
8530 PPINTSP$;CHR$(18);LEFT$(H$,
7);CHR$(178);RIGHT$(H$,15)
8540 PPINT"    ";CHR$(153);"<UP>
 ";CHR$(155);CHP$(18);CHR$(125);
"POS ";
8550 PRINTCHP$(146);"  ";CHR$(18
);CHR$(125);CHP$(146);"
      ";CHR$(18);
8560 PRINTCHR$(125);CHR$(146),CH
P$(153),"<DOWN>",CHR$(155)
8570 PRINTSP$;CHP$(18);LEFT$(E$,
7);CHP$(177);RIGHT$(E$,15):PRINT
8580 PPINTSP$;LEFT$(H$,10);CHR$(
178);RIGHT$(H$,12)
8590 PRINTSP$;CHR$(125);SP$;CHR$
(125);SP$;"  ";CHR$(125)
8600 PPINTSP$;LEFT$(B$,10);CHR$(
177);RIGHT$(B$,13):PRINT
8610 PRINTSP$;CHR$(18);"
                "
8620 PRINTSP$;CHP$(18);" ";CHP$(
158);"1 TO CHANGE PRESS [C]";CHR
$(155);" "
8630 PRINTSP$;CHR$(18);" ";CHR$(
158);"2 USE RETURN TO ENTER";CHR
$(155);" "
8640 PRINTSP$;CHR$(18);" ";CHR$(
158);"3 CHANGE ALL YEAR [A]";CHR
$(155);" "
8650 PRINTSP$;CHR$(18);" ";CHP$(
158);"4 TO QUIT PRESS [M]  ";CHP
$(155);" "
8660 PRINTSP$;CHR$(18);"
                ";CHR$(19);:GOTO8
350
9000 REM MONTHLY PREVIEW ON SCRE
EN
9010 SM=PR:PRINTCHR$(147);CHP$(1
53);"PRESS     ";CHR$(18);CHP$(15
5);
9020 PRINT"INCOME     "CHR$(5);M
O$(SM);CHR$(155);"          "
9030 FORP=1TOI
9040 IFI(SM,P)=OTHEN9080
9050 PRINTSP$;CHR$(18);CHR$(155)
;MID$(STR$(P)+"  ",2,3);
9060 PRINTCHR$(18);I$(P);" ";CHR
$(146);CHP$(5);
9070 PRINTRIGHT$("     "+STP$(I(
SM,P)),5);CHR$(155);CHR$(18);" "
9080 NEXTP
9090 PRINTCHR$(153);"I,M,P,Y, ";
CHR$(155);CHR$(18),"TOTAL INCOME
   =";CHR$(5);
9100 PRINTCHP$(18);RIGHT$("
"+STR$(IT(SM)),5);CHR$(155);" "
9110 PRINTCHR$(153);"E OR I.  ";
CHR$(158);CHR$(18);"EXPENSES
          "
9120 FORP=1TOE
9130 IFE(SM,P)=OTHEN9170
9140 PRINTSP$;CHR$(158),CHR$(18)
;MID$(STR$(P)+"  ",2,3);
9150 PRINTE$(P);" ";CHR$(146);CH
R$(5);
9160 PRINTRIGHT$("     "+STP$(E(
SM,P)),5);CHR$(158);CHR$(18);" "
9170 NEXTP
9180 PRINTSP$;CHP$(158);CHR$(18)
;"TOTAL EXPENSES =";CHP$(5);
9190 PRINTRIGHT$("     "+STP$(ET
(SM)),5);CHR$(158);" "
9200 PRINTSP$;CHP$(18),CHP$(159)
;"BALANCE          ";
9210 PRINTCHP$(18);RIGHT$("
"+STP$(MT(SM)),5);CHP$(159);" ";
9220 PRINTCHP$(155);CHP$(19)
9230 A$=""
9240 GETKEYA$
9250 IFA$="I"THENBB$="I":GOTO8CO
0
9270 IFA$="E"THENBB$="E":GOTO8CO
0
9280 IFA$="M"THENSM=PR:GOTO4000
9290 IFA$="Y"THENSM=PR:GOTO3000
9300 IFA$="P"THENSM=PR:GOTO5000
9310 PPINTCHP$(147);CHP$(153);"P
RESS     ";CHP$(155);CHP$(18);
9320 SM=SM+1.IFSM=13THENSM=1
9330 GOTO5020
10000 REM EXPENSE AND INCOME ANN
UAL ANALYSIS
10010 SM=PR:PRINTCHR$(147):PRINI
:PRINT:PRINT
10020 PRINTSP$;CHP$(18);CHR$(159
);" ANALYSIS OF EXPENSES "
10030 PRINT
10040 PPINTSP$;CHP$(18);" OR INC
OME? ";CHP$(130),"(E OR I) ";CH
R$(5);CHP$(132)
10050 GETKEYD$
10060 IFD$="I"THEN10080
10070 IFD$="E"THEN10100
10080 GOTO4000
10090 PRINT:PPINT:PPINTSP$;"INCO
ME NUMBER FOR ANALYSIS.";GOTO101
10
10100 PRINT:PRINT:PRINISP$;"EXPE
NSE NUMBER FOR ANALYSIS."
10110 F=1:PRINT:INPUT"          "
;F:PRINTCHR$(147)
10120 IFF>IORF<1THENPRINTCHR$(19
):GOTO10060
10130 IFF>IANDD$="I"THENPRINTCHR
$(19):GOTO10060
10140 PRINTCHR$(19);SP$;CHR$(5);
"ANALYSIS OF"
10150 PRINT
10160 XX=0
10170 PRINTSP$;
10180 IFD$="I"THENPRINTCHP$(18);
I$(F);CHR$(146);CHR$(153);F;CHR$
(157);" "
10190 IFD$="E"THENPRINTCHR$(18);
E$(F);CHR$(146);CHR$(153);F;CHR$
(157);" "
10200 PRINT
10210 FORP=1TO12
10220 PRINTSP$;CHP$(18);CHR$(159
);MO$(SM);CHR$(146);CHR$(5);"
 ";
10230 IFD$="I"THEND=I(SM,F)
10240 IFD$="E"THEND=E(SM,F)
10250 XX=XX+D
10260 PRINTRIGHT$("      "+STR$(D
),5)
10270 SM=SM+1:IFSM=13THENSM=1
10280 NEXTP
10290 PRINT:PRINTSP$;"==========
=="
10300 PRINTSP$;"TOTAL  ";
10310 PRINTRIGHT$("      "+STP$(X
X),5)
10320 PRINTSP$;"-----------":PR
INT
10330 PRINTSP$;CHR$(153);"CURSOR
  <UP> OR <DOWN>":PRINT
10340 PRINTSP$;CHR$(5);"ANALYSIS
 OR MENU? (A/M)"
10350 GETKEYE$
10360 IFE$=CHR$(145)ANDF>1THENF=
F-1:GOTO10120
10370 IFE$="M"THEN4000
10380 IFE$="A"THEN10000
10390 IFE$=CHR$(17)ANDD$="I"ANDF
<IANDI$(F+1)<>"                "THEN
F=F+1:GOTO10120
10400 IFE$=CHR$(17)ANDD$="E"ANDF
<EANDE$(F+1)<>"                "THEN
F=F+1:GOTO10120
10410 GOTO10350
11000 REM SAVE FILE TO DISK OR T
APE
11010 PRINTCHR$(147):IFY=OOR2=OT
HEN4000
11020 PRINTCHR$(147);CHR$(155);C
HR$(130):FORP=1TO6:PRINT:NEXTP:P
$=""
11030 PRINT"     SAVE ON TAPE OR
 DISK? (T OR D)":PRINTCHR$(132):
GETKEYP$
11040 IFP$="T"ORP$="D"THEN11050:
ELSE4000
11050 PRINTCHP$(147);CHR$(5):FOR
P=1TO5:PRINT:NEXTP:PRINTSP$;"
 ";
11060 PRINTCHR$(130);CHR$(18);"
SAVING FILE. ";CHR$(132);CHR$(14
6)
11070 IFP$="D"THENOPEN1,8,2,"@0:
EXPENSES,S,W":ELSEOPEN1,1,1,"EXP
ENSES"
11080 FORC=1TO12:FORP=1TO5:PRINT
#1,E(C,P):NEXTR:NEXTC:CLOSE1
11090 IFP$="D"THENOPEN2,8,2,"@0:
INCOME,S,W":ELSEOPEN2,1,1,"INCOM
E"
11100 FORC=1TO12:FORP=1TO1:PRINT
#2,I(C,R):NEXTR:NEXTC:CLOSE2
11110 GOTO4000
12000 REM LOAD FILE FROM DISK OR
 TAPE
12010 PRINTCHR$(147);CHR$(155);F
OPP=1TO6:PRINT:NEXTP:PRINT"   ";
12020 P$="":PPINT"LOAD FROM TAPE
 OR DISK? (T OR D)":GETKEYP$
12030 IFP$="D"ORP$="T"THEN12040:
ELSE4000
12040 PRINTCHR$(147):FORP=1TO6:P
RINT:NEXTP:PRINISP$;"    ";
12050 PRINTCHR$(130);CHR$(5);CHR
$(18);" LOADING FILE. ";CHR$(132
);CHR$(146)
12060 IFP$="D"THENOPEN1,8,2,"0:E
XPENSES,S,R":ELSEOPEN1,1,0,"EXPE
NSES"
12070 FORC=1TO12:FORP=1TO5:INPUT
#1,E(C,P):NEXTR:NEXTC:CLOSE1
12080 IFP$="D"THENOPEN2,8,2,"0:I
NCOME,S,R":ELSEOPEN2,1,0,"INCOME
"
12090 FORC=1TO12:FORP=1TO1:INPUT
#2,I(C,P):NEXTR:NEXTC:CLOSE2
12100 KK=1.PRINTCHR$(147);CHP$(5
):FORP=1TO5:PRINT:NEXTP:PPINT"
          ";
12110 PRINTCHR$(19),CHR$(130);"
CALCULATING. ";CHR$(132);CHR$(14
6):GOTO2000
```

Bomb Jack

Bomb Jack was a number one chart-topping smash hit when it was originally released by Elite but now it is back as part of Elite's new Encore budget range. This 1985 Tecmo coin-op conversion would still rank highly among todays releases as it combines all the essential elements of a great game. It's undoubtedly addictive as it seems to burn up hours while you sit at the keyboard hooked on it's simple but challenging gameplay.

As the game opens you control the Bomb Jack, a rather agile character that can leap and around the platforms that are scattered around the screen. While in the background the sphinx watches your every move. You job is to collect, and disarm, all the bombs that fill the screen if you succeed you'll go on to the next, more challenging screen.

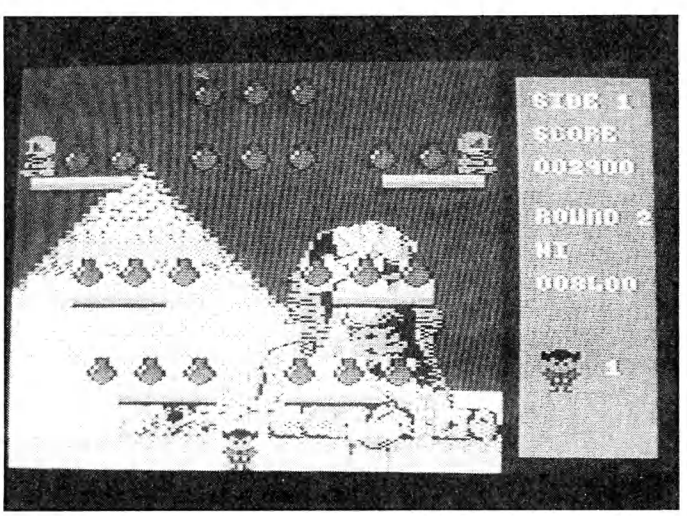
Unfortunately, the platforms are patrolled by aliens that walk along the platforms. However, during the game they will drop down to a lower platform, patrol that for a while, and so on until they reach the bottom of the screen where they mutate into giant, rotating balls that home in on you.



Giant birds of prey also hunt you down from the onset of your quest and will pursue you relentlessly until you either clear the screen of bombs or are caught and lose one of your lives. Collisions with any of these nasties is fatal and so you will have to keep your wits about you if you are going to survive. Also you must act quickly as soon as the patrolling aliens mutate you're going to be surrounded in homing enemies and you'll be left with nowhere to go.

The secret of Bomb Jack lies in control of the character as you can walk left and right and jump and maintain a limited control over your flight through the air allowing you to clear a whole group of bombs in a single leap. It is important to remember that you can only jump and can't fly as if you make this mistake you'll suddenly find yourself in an uncontrolled plummet towards oblivion and there will be nothing you can do about it.

Soon you'll begin to learn the best route around the screen that will take you onto to platforms such before or after the patrolling aliens arrive. You'll still have to lead the flying brids astray by sending them the long way around which should give you time to clear the bombs and progress to the next level.



As you meet the challenge offered by each level you'll have to contend with more complex patterns of platforms that limit your movement more and more as you go through the game. This means you have to stay even more alert as it will become more difficult to keep the birds at bay especially since the number of these begins to increase.

This is an excellent conversion of the coin-op original that hasn't lost any of its fiendish gameplay although some character smudging may distract from the appearance of the game.

Touchline:

Title: Bomb Jack. Supplier: Encore (Elite), Eastern Ave., Lichfield, West Midlands, WS13 6RX. TEL: 0543 414885. Price: £1.99.

Windows +4 / C16

Professionalise your programs with this handy window routine.

By D. Milne

There have been many programs for the C64 to emulate the window feature of machines like the IBM PC but none for the C16/PLUS4 despite these machines having a simple window-like feature built in already.

This program for the C16/PLUS4 gives these machines IBM PC style windows. It consists of three machine code routines designed to be used from BASIC.

Entering the program

Type in and save the BASIC boot program in listing 1. If your are using tape then change the DLOAD to a LOAD. Next type in and save the BASIC loader WINDOW.BAS given in listing 2.

To use the program, reset the computer, load and run the boot program (if using tape then do not press stop after it has loaded). This will load and run the BASIC loader and after a slight delay the cursor will reappear and the routines are ready to use.

Using the routines from BASIC

The first step here is to set up the pointer to the storage stack and protect the stack from BASIC. It is best to put the stack at the top of memory and lower the top of memory pointers in 51,52 and 55,56. For instance if we wished the stack to start at $E000 then we would poke 255,223 into 51,52 and 55,56, preform a CLR then poke 0,224 into the stack pointer in 4,5. (See the demonstration program lines 50 and 60 as a further example.)

Next, the parameters for the windows must be calculated. The four parameters are the number of columns (COL), the number of rows (ROW), the position of the window's top left corner in screen RAM ((MEMLOC) and the colour of the border (CLR). The number of columns and rows is self explanatory except to note that this does not include the border, only the actual window.

The position of the top left corner in screen RAM is the memory location where the top left corner of the border is and can be found from the screen memory map in the user guide. The colour is calculated from the values used in the COLOR statement and is calculated as follows:

value = colour #-1 + 16* luminance #

These parameters are used in a SYS statement of the form SYS 4633,COL,ROW, MEMLOC,CLR to set up a window on the screen. Removing a window is a complished by SYS 4253 followed by PRINT'(HOME2)' if that was the last window on the screen or SYS 4648,COL,ROW,MEMLOC,CLR if there are other windows where the parameters used refer to the window you wish to use not the window just removed.

Note that the window removed is the last one placed on the screen. Note also that the routine at 4648 can be used by itself to set up C16/PLUS4 pseudo-text-windows as described in the sure guide without mucking around with ESC codes and cursor key codes.

Writing to the window is done using the PRINT statement as normal. The windows can also be scrolled up and down using the appropriate ESC codes such as PRINT CHR$(27)+'v'; which scrolls the screen up.

Important

These routines contain no error trapping, so it is up to the user to ensure that the parameters given do not cause the window to wrap-round or to disappear off the screen as this will cause problems such as the overwriting of the window routines themselves! You should also avoid using PRINT'(HOME2)' or the CHAR statement as these remove the C16/PLUS4 pseudo-text-window although these can always be reset using the routine at 4648.

Using the routines from machine code

Listing 3 is an assembly language lising of the routines. The important entry points for machine code are:

PROGRAM: WINDOW.BAS

Listing 2

```
10 REM PLUS/4 WINDOWS
20 REM BASIC LOADER
30 FORA=4096TO4654
40 READB:POKEA,B
50 NEXT
60 DATA120,141,63,255,32,87,16,2
30,208,166,209,232,232,164,208,1
77
70 DATA2,132,6,160,0,145,4,164,6
,32,96,16,132,6,160,1
80 DATA145,4,164,6,32,115,16,136
,192,255,208,227,32,129,16,202
90 DATA208,219,32,143,16,160,0,1
65,2,145,4,200,165,3,145,4
100 DATA200,165,209,145,4,200,19
8,208,165,208,145,4,32,115,16,32
110 DATA115,16,141,62,255,88,96,
165,210,133,2,165,211,133,3,96
120 DATA165,3,56,233,4,133,3,177
,2,72,165,3,24,105,4,133
130 DATA3,104,96,165,4,24,105,2,
133,4,165,5,105,0,133,5
140 DATA96,165,2,24,105,40,133,2
,165,3,105,0,133,3,96,165
150 DATA2,56,233,40,133,2,165,3,
233,0,133,3,96,120,141,63
160 DATA255,32,238,16,32,238,16,
160,0,177,4,133,2,200,177,4
170 DATA133,3,200,177,4,133,209,
200,177,4,133,208,230,208,230,20
8
180 DATA166,209,232,232,160,0,32
,238,16,132,6,160,0,177,4,164
190 DATA6,145,2,132,6,160,1,177,
4,164,6,32,252,16,200,196
200 DATA208,208,227,32,143,16,20
2,208,219,141,62,255,88,96,165,4
210 DATA56,233,2,133,4,165,5,233
,0,133,5,96,72,165,3,56
220 DATA233,4,133,3,104,145,2,16
5,3,24,105,4,133,3,96,32
230 DATA87,16,164,208,200,169,86
,145,2,32,88,17,136,192,255,208
240 DATA246,166,209,32,129,16,16
4,208,200,169,86,145,2,32,88,17
250 DATA136,169,32,145,2,32,88,1
7,136,208,248,169,86,145,2,32
260 DATA88,17,202,208,222,32,129
,16,164,208,200,169,86,145,2,32
270 DATA88,17,136,192,255,208,24
6,96,72,165,3,56,233,4,133,3
280 DATA165,212,145,2,165,3,24,1
05,4,133,3,104,96,165,211,56
290 DATA233,12,133,211,162,0,165
,211,208,6,165,210,201,40,144,17
300 DATA165,210,56,233,40,133,21
0,165,211,233,0,133,211,232,24,1
44
310 DATA229,164,210,232,200,169,
19,32,210,255,169,19,32,210,255,
192
320 DATA0,240,9,169,29,32,210,25
5,136,24,144,243,224,0,240,9
330 DATA169,17,32,210,255,202,24
,144,243,169,27,32,210,255,169,8
4
340 DATA32,210,255,198,209,198,2
08,166,208,224,0,240,9,169,29,32
350 DATA210,255,202,24,144,243,1
66,209,224,0,240,9,169,17,32,210
360 DATA255,202,24,144,243,169,2
7,32,210,255,169,66,32,210,255,1
69
370 DATA19,76,210,255,32,11,18,1
32,208,32,11,18,132,209,32,11
380 DATA18,132,210,133,211,32,11
,18,132,212,96,32,145,148,32,44
390 DATA147,32,238,157,165,21,16
4,20,96,32,244,17,32,0,16,32
400 DATA87,16,32,15,17,76,109,17
,32,244,17,76,109,17,78,89
```

(1) $1000 : save screen area
(2) $109D : restore screen area
(3) $110F : draw window
(4) $116D : set up pseudo-text-window

The parameters are held as follows:

$D0 : number of columns
$D1 : number of rows
$D2,D3 : position of top left corner on screen
$D4 : colour of border

PROGRAM: WINDOWS AUTOBOOT

```
10 POKE43,49:POKE44,18:POKE45,51
:POKE46,18
20 POKE47,58:POKE48,18:POKE4656,
0:CLR
30 PRINT"[CLR]DLOAD"+CHR$(34)+"W
INDOW.BAS"
40 PRINT"[DOWN4]RUN"ROR
50 POKE1319,19:POKE1320,13:POKE1
321,13:POKE239,3
60 NEW
```

Listing 1

Routines (1) and (4) require all but the colour parameter, routine (3) requires all the parameters and routine (2) requires no parameters.

And finally...

A demonstration program is included (listing 4) to show how these routines can be used.

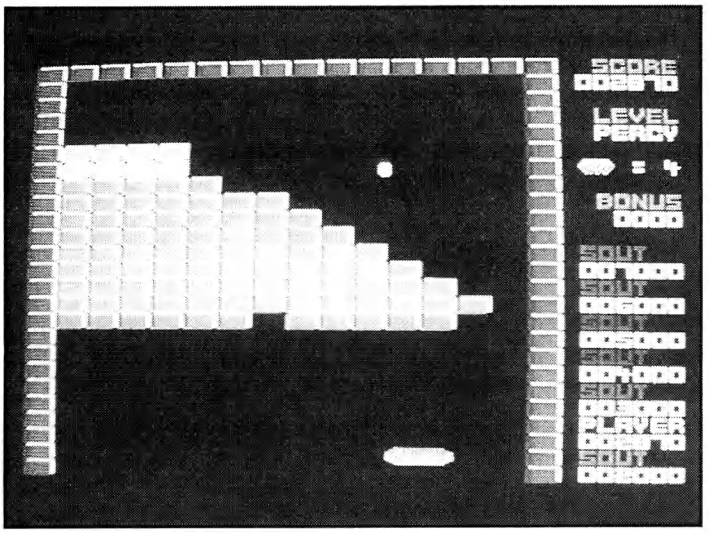
If you wish a different character for the border then change the 86 in lises 230,240,250 and 260 of the BASIC loader (listing 2) to whichever screen code is required.

These routines could be used in programs using pull down menues, help windows etc. Be imaginitive and experiment.

PROGRAM: WINDOWS DEMO

```
10 REM PLUS/4 WINDOWS
20 REM DEMONSTRATION
30 REM PROGRAM V1.1
40 :
50 POKE51,255:POKE52,226:POKE55,
255:POKE56,226:CLR
60 POKE4,0:POKE5,227
70 PRINT"[CLR]CBM PLUS/4 WINDOWS
"
80 PRINT"[DOWN7]NORMAL SCREEN AR
EA."
90 :
100 REM SET UP WINDOW PARAMETERS
110 :
120 AD=32763:DIMW(4,3)
130 FORA=0TO4:FORB=0TO3:READW(A,
B):NEXTB,A
140 DATA38,22,3112,33,20,10,3196
,98,4,10,3278,67,10,4,3575,84,6,
2,3733,50
150 :
160 REM SET UP WINDOWS
170 :
180 N=0:GOSUB490
190 PRINT"   THIS IS A DEMONSTRA
TION OF IBM     STYLE WINDOWS ON
 THE COMMODORE PLUS/4."
200 PRINT"THE DEMONSTRATION PROG
RAM WILL USE    FIVE WINDOWS INC
LUDING THIS ONE TO    ";
210 PRINT"SHOW HOW THE WINDOWS A
RE USED IN      PRACTICE."
220 PRINT"   THESE WINDOWS WILL
BE PLACED ON TO THE SCREEN THEN
SLOWLY REMOVED."
230 PRINT"PRESS A KEY."
240 GETKEYA$
250 N=1:GOSUB490
260 PRINT"  THIS IS WINDOW    NU
MBER 1 AND IS 20  COLUMNS WIDE."
270 GOSUB560:GOSUB560:N=2:GOSUB4
90
280 PRINT"LOADSAVELISTHELPCAT EN
D"
290 GOSUB560
300 N=3:GOSUB490:PRINT"ALSO OVER
 HERE!"
310 GOSUB560
320 N=4:GOSUB490:PRINT"SMALL"
330 GOSUB560:GOSUB560:GOSUB560
340 :
350 REM REMOVE WINDOWS
360 :
370 SYS4253:N=3:GOSUB540
380 PRINT"[DOWN4]BACK HERE.":GOS
UB560
390 SYS4253:N=2:GOSUB540:PRINT"[
DOWN15]":GOSUB560
400 SYS4253:N=1:GOSUB540:PRINT"[
CLR]NOW WE ARE HERE.":GOSUB560
410 SYS4253:N=0:GOSUB540:GOSUB56
0
420 PRINT"[CLR,SPC3]FINALLY WE R
EMOVE THIS WINDOW."
430 GOSUB560:SYS4253:GOSUB560
440 PRINT"[HOME,DOWN17]"
450 END
460 :
470 REM SET UP WINDOW # N
480 :
490 SYS4633,W(N,0),W(N,1),W(N,2)
,W(N,3)
500 RETURN
510 :
520 REM RESET UP WINDOW # N
530 :
540 SYS4648,W(N,0),W(N,1),W(N,2)
,W(N,3)
550 RETURN
560 FORD=1TO1000:NEXT:RETURN
```

Listing 3



Arthur Noid

Arcade games come and go but few have had such consistent appeal as the Breakout style of games. First there was the original Breakout which first appeared in the arcades over ten years ago which was quickly followed by Super Breakout that featured double bats and trapped balls that you could release to rack up the high scores. By now the home computers were carrying versions of Breakout in a variety of guises.

Two years ago Breakout made a comeback in the arcades as Arkanoid which generated a whole new era of Breakout fever. Arthur Noid is the best version you can find on the C16/Plus/4.

Arthur Noid was released only recently by Alternative Software who is a relative newcomer to the budget market but has made an outstanding start and is sure to continue if it releases games of this quality.

Arthur Noid is obviously "inspired" by Arkanoid but has added a few touches of it's own to make it an incredibly addictive game. Games reviewers often talk about games that drive you back, time and time again, for just one more game and this definitely belongs to that unique category.

A total of 32 main levels wait to challenge you that consist of increasingly difficult patterns of bricks. Your job is to destroy them all by hitting a ball against them by controlling a keyboard or joystick operated bat that can be moved across the bottom of the screen. Each screen consists of read, yellow, blue and green bricks that can be destroyed by a single hit as well as grey and gold ones.

Gold bricks are indestructible and form the boundaries of the screen as well as add to your problems by fencing in bricks you must destroy leaving you a difficult angle to hit but once you find it the ball bounces around the gold bricks and wipes them out.

Grey bricks can be destroyed, and must be if you are to complete the level, but must be hit two, three or four times depending on the game level. This actually has it's advantages as if you manage to break through a wall a row of grey bricks will keep it behind the wall for longer so that it clears out the wall from behind leaving you to pick and choose from the bonus barrels.

Your ability to pick off the right bonus barrel at the right time will decide how far up the high score table you will climb. These appear at random and roll down the screen until either you collect them by hitting them with your bat or they disappear, out of reach, off the bottom of the screen.

Catching a blue barrel increases the size of your bat by more than double, a cyan barrel makes the ball stick to the bat for a few seconds to allow you to aim it, a yellow one slows the ball down, green gives you an extra life, purple splits your ball into three and red mutates your bat into a twin firing laser bat that can blast away at the bricks.

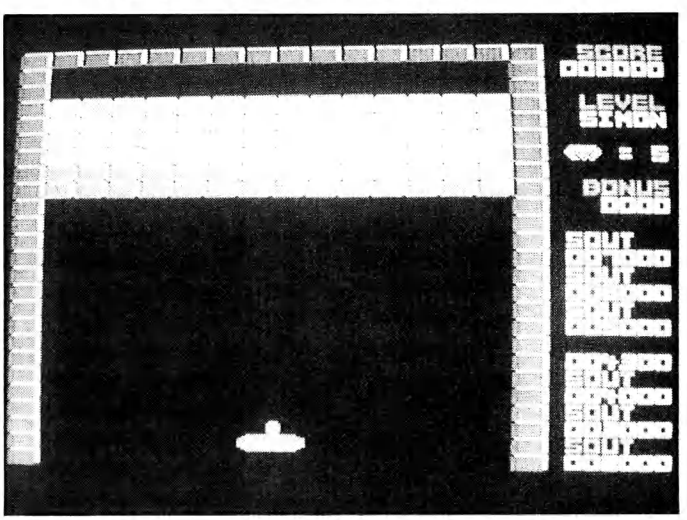
Once you've got the bonus barrel that you think you'll need to clear the screen you then should avoid that other barrels or they'll take effect.

To add to your problems aliens float around the screen. They don't kill you but they do collide with the ball and deflect it. This usually isn't a problem but can be deadly if it happens just in front of your bat.

To the right of the main display your score is constantly updated and added into the high score table so you can see how you're moving up the order as you clear the levels.

One of the best features of the game are the bonus screens that appear between the main levels and challenge you to break through a wall an inch from your bat or clock up the most bounces within a time limit.

A superb version of an arcade legend.



Touchline:

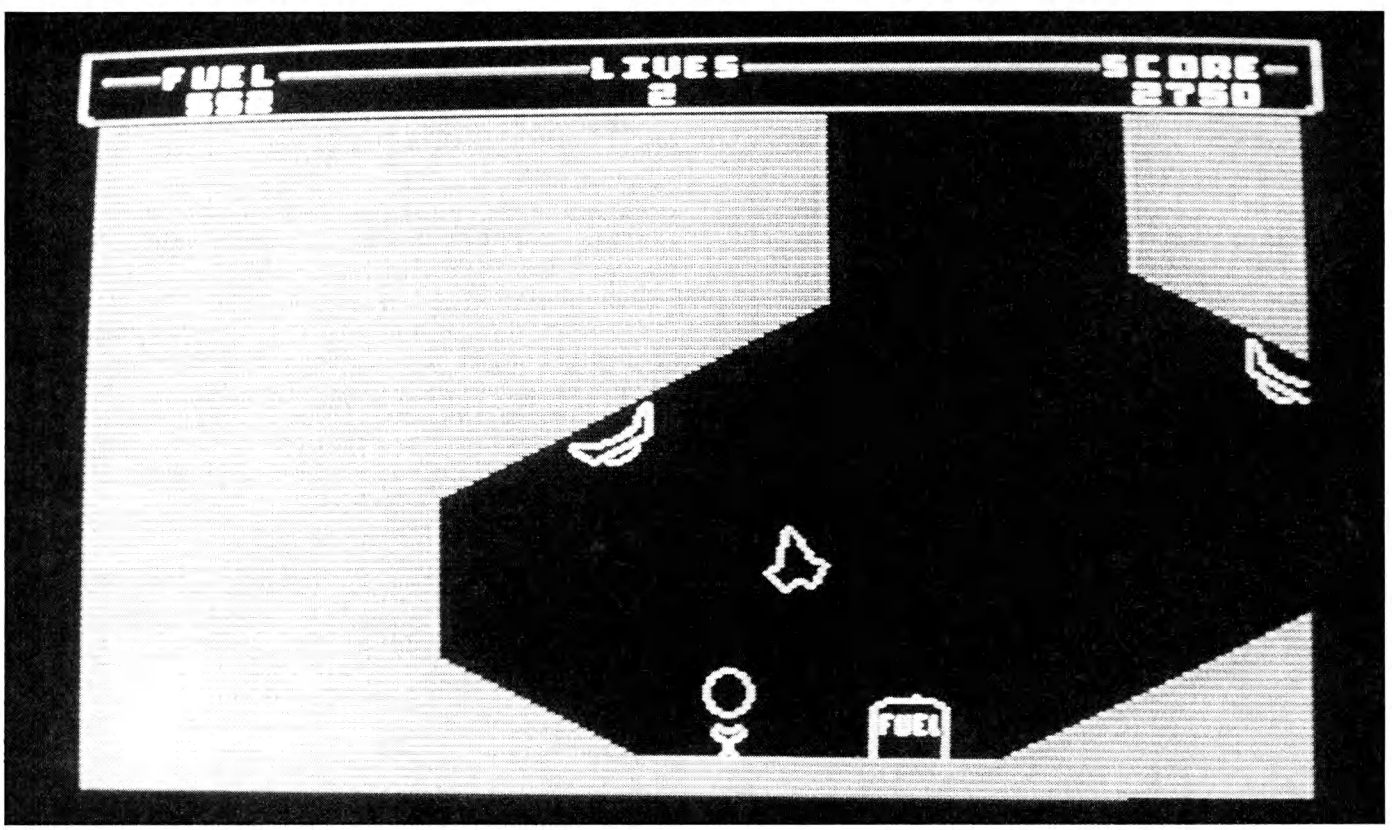
Title: Arthur Noid. Supplier: Alternative Software, Units 3-6 Baileygate Industrial Estate, Pontefract, West Yorkshire, WF8 2LN. TEL: 0977 797777.
Price: £1.99.

Thrust

This is one of the budget games that staked the claim that cut price software was just as good as it's full priced counterparts.

Although it's not actually a coin-op conversion Thrust has all the components that would make it a hit in the arcades. It's easy to learn and almost impossible to master. Having said that, it's just as well that you don't have to pay 20p for a game or it could prove very expensive.

In the game you play a resistance fighter who must brave the Empire's storage planets to find the Klyston power pods to drive the captured Battle-grade starships that will spearhead the resistance attack.



These power pods are guarded by limpet guns that cling to rock faces and under hanging edges waiting for anything to come into their sights and your ship is the most likely target. The guns are powered by nuclear reactors that are usually on the planets surface. By hitting them a few times with your laser bolts you may be able to silence the guns for a valuable seconds. However if you blast them too much then they'll go critical and explode destroying you, the limpet guns and the planet.

The alternative is to shoot it out with guns which may be unavoidable on the higher levels where they are buried in caverns deep below the planets surface. They take only one hit to destroy them and your shield may save you.

The ship you control looks like a refugee from that classic arcade game Asteroids and is controlled by keys that rotate it right and left and thrust which moves it forward. It also has a tractor beam by which it grabs and holds onto the power pods if it gets close enough to them.

The first level is simple as there is a reactor, one limpet gun, the power pod and a fuel store (that you can drain using your tractor beam) all on the planets surface. This allows you time to practise your control over the ship that must become accurate to a degree where you can thrust into a cavern, hover take out the limpet guns, descend to top up on fuel and pick up the pod and then thrust out. If that sounds impossible then give in now because that's what you have to do in level two!

In later levels the pods are buried deep underground in caverns linked with narrow, ship smashing, corridors and you'll even have to deal with planets with reverse gravity and batteries of limpet guns.

Fuel is essential in this game as it is used up every time you press the thrust button so you must keep you moves to a minimum and take advantage of every fuel dump you find.

The result is a superb game that will take a player with a steady hand and steal nerve to conquer it.

Touchline:
Title: Thrust. Supplier: Firebird, 64/76 New Oxford Street, London WC1A 1PS.
Price: £1.99.

Datafile

For the Plus/4 and C16
Uses Datasette and printer

By Nigel Smith

If like me you bought a Commodore Plus/4 and then realised that you couldn't use the built in database without a disk drive, or you own a C16 and you need one then this program is for you. It allows you to maintain a database on tape and perform a variety of operations on it. For example you could store names and addresses on it or store your entire record collection, and at the press of a key you can sort it, add up the contents of certain fields or search for data throughout the file.

Typing It In

It should be typed in exactly as shown. I have included lots of REMs and colons to space out the listing and to make it easier to follow but if you own a C16 you might be as well removing them. Alternatively you can use the Cruncher program which was printed in the July 1987 issue of Your Commodore to create more space and speed it up a little. There are also some lines which contain two HOME functions. Both these should be typed in as they reset any windows which may have been created.

Instructions

When you first run the program there will be no records in memory so you have the choice of loading a file or creating a new one. If you choose the LOAD FILE option you will be asked for a filename. If you want to load in the first file on tape just press return without entering anything. The file will then load and you will be presented with menu 2.

If you select the CREATE FILE option then you will be asked for a filename. This is the name that your file will be saved under. You will then be asked how many fields each record is to contain, you can have up to nine fields in each record and each fieldname can be up to ten characters long. You then enter all your fieldnames. It will then ask how many records you wish to enter, up to 500. Now begins the tedious task of entering all the data. Once this is done you will be presented with the second menu. If you make any mistakes, don't worry as you will have a chance to correct them later.

Here are the commands in menu 2:

A. Add a record
B. Insert a record
C. Delete a record
D. Edit a record
E. View records
F. Search for word
G. Sort file
H. Add up contents of a field
I. Print records
J. Save file
K. Load file
L. Create new file

A. Add a record

This adds a record on to the end of the file. It is useful for adding on a lot of records. Just enter the data as before.

B. Insert a record

If you only need to add a few records and the file has been sorted then this will allow you to insert them anywhere so that you don't have to spend time resorting the file.

C. Delete a record

This allows you to delete any record you no longer need. Just enter the record number and confirm your choice.

D. Edit a record

This is when you get the chance to correct any mistakes you made earlier. Enter the record number and go through each field in turn. If you don't wish to alter that field just press return otherwise use the normal cursor functions and edit the mistake.

E. View records

When you select this option the first record in the file will be displayed on screen. Use the left and right cursor keys to flick through the records or press G to goto a specific record. Pressing E will take you back to the menu.

F. Search for word

This allows you to search through the entire file for occurencies of a particular word. If a match is found in a record then that record will be displayed on screen. You will then be asked if you wish to continue searching. If you type Y then it will

continue, otherwise you will be returned to the menu. The program doesn't distinguish between upper case and lower case characters, so Hello, hello and HELLO will all match.

G. Sort file

This option sorts your file into alphabetical order by any field. Just enter the number of the field you wish to sort by and wait. Depending on how many records there is in the file it may take a long time.

H. Add up contents of a field

If you have a file which contains numbers, prices etc, then you may want to know the total in a particular field. Just choose the field to add up and it will tell you the total.

I. Print records

This will let you print all or some of the records to an MPS 801 or MPS 803. If you select PRINT SOME RECORDS then you will be asked to enter a record number. This record will appear on screen and if you want it printing press Y. Otherwise press N.

K. Load file

Instructions for this were printed earlier.

J. Save file

This will save your file under the name you typed in earlier.

L. Create new file

This will allow you to restart and make a new file as before. If you select LOAD FILE or CREATE NEW FILE you will be asked to confirm your choice as it will wipe the current data.

Notes

C16 users should change the maximum number of records at lines 240 and 2360 to MAX=50 instead of MAX=500. If you need any more records you can store them in two or more different files.

If you need a numerical sort then I suggest adding leading zeros to any numbers. E.g 45, 123 and 734 would be sorted as 123,45 and 734 which is obviously wrong, so use the numbers like this – 045,123,734 and they will be sorted correctly.

```
PROGRAM: DATAFILE



10 rem ********************
********
20 rem * plus 4/c16 datafil
e v5.5 *
25 rem *      by nigel smith
       *
30 rem * c16 users may dele
te all *
40 rem * rems and colons to
 make  *
50 rem *       more space
       *
60 rem *--------------------
-------*
70 rem *        uses 6.2k
       *
80 rem ********************
********
90 color4,6,5:color0,2,7
100 printchr$(14);chr$(8);
110 rem ** first menu **
120 print"[HOME2,CLS,ORNG,D
OWN,SPC10]Plus 4/C16 Datafi
le"
130 print"[SPC10,CT19]"
140 print"[DOWN,GREEN,SPC13
]First Menu"
150 print"[SPC13,CT10]"
160 print"[DOWN,DBLU,SPC6]A
.[RED]Create new file."
170 print"[DBLU,SPC6]B.[RED
]Load a file"
180 print"[DOWN11,PURPLE] P
ress Selection:[DOWN,CT,UP,
LEFT,BLACK]";
190 c$="ab":gosub760
200 ifc=2thengosub2620
210 :
220 rem * create file *
230 print"[HOME2,CLR,RED,SP
C12,RVSON] CREATE FILE "
240 gosub820:clr:max=500
250 input"[DOWN,DBLU]Enter
Filename :[GREEN]";f1$
260 iflen(f1$)>16then250
270 input"[DOWN,DBLU]Enter
number of fields (1-9) :[GR
EEN]";nf
280 ifnf<1ornf>9then270
290 dim f1$(max,nf)
300 forf=1tonf
310 print"[DOWN,DBLU]Enter
field name";f;":[GREEN]";:i
nputf1$(0,f)
320 iflen(f1$(0,f))>10then3
10
330 f1$(0,f)=f1$(0,f)+left$
("          ",10-len(f1$(0,
f)))
340 nextf
350 input"[DOWN,DBLU]Enter
number of records to start
:[GREEN]";nr
360 if nr<1ornr>maxthen350
370 print"[DOWN,BRN]Is all
the above data correct (y/n
)?"
380 getkeya$:ifa$="n"thengo
to230
390 :
400 rem * input data *
410 print"[HOME2,CLR,RED,SP
C12,RVSON] INPUT DATA "
420 gosub820
430 forr=1tonr
440 print"[DOWN,PURPLE]Reco
rd number";r;"[DOWN]"
450 forf=1tonf
460 print"[DBLU]";f;f1$(0,f
);":[GREEN]";:inputf1$(r,f)
470 nextf,r
480 :
490 rem ** second menu **
500 print"[HOME2,CLR,ORNG,D
OWN,SPC10]Plus 4/C16 Datafi
le"
510 print"[SPC10,CT19]"
520 print"[DOWN,GREEN,SPC16
]Menu"
530 print"[SPC16,CT4]"
540 print"[DOWN,DBLU,SPC6]A
.[RED]Add a record."
550 print"[DBLU,SPC6]B.[RED
]Insert a record."
560 print"[DBLU,SPC6]C.[RED
]Delete a record."
570 print"[DBLU,SPC6]D.[RED
]Edit a record."
580 print"[DBLU,SPC6]E.[RED
]View records."
590 print"[DBLU,SPC6]F.[RED
]Search for word."
600 print"[DBLU,SPC6]G.[RED
]Sort file."
610 print"[DBLU,SPC6]H.[RED
]Add up contents of a field
."
620 print"[DBLU,SPC6]I.[RED
]Print records."
630 print"[DBLU,SPC6]J.[RED
]Save file to tape."
640 print"[DBLU,SPC6]K.[RED
]Load file from tape."
650 print"[DBLU,SPC6]L.[RED
]Create new file."
660 print"[DOWN,PURPLE] Pres
s Selection:[DOWN,CT,UP,LE
FT,BLACK]";
670 c$="abcdefghijkl":gosub
760
680 ifc<>11andc<>12then720
690 print:print"[RED]ARE YO
U SURE (Y/N)?"
700 getkeya$:ifa$<>"y"then5
00
710 ifc=12then230:elsegosub
2620:goto500
720 oncgosub850,970,1180,13
90,1520,2020,1710,2320,2770
,2480
730 goto 500
740 :
750 rem * get a key routine
 *
760 getkeyk$
770 ifk$=chr$(13)andc<>0the
nreturn
780 ifk$=chr$(20)andc<>0the
nc=0:print" [LEFT]";:goto76
0
790 c=instr(c$,k$):ifc=0the
n760
800 printchr$(asc(k$)+32);"
[LEFT]";:goto760
810 rem * protect top of sc
reen *
820 print"[HOME2,DOWN]";chr
$(27);"t";:return
830 :
840 rem * add a record *
850 print"[HOME2,CLR,RED,SP
C12,RVSON] ADD A RECORD "
860 gosub820
870 ifnr=maxthenprint"[BLAC
K,DOWN,SPC8,FLASHON]FILE FU
LL [FLASHOFF]- PRESS A KEY"
:getkeya$:return
880 nr=nr+1
890 print"[DOWN,PURPLE]Reco
rd number";nr;"[DOWN]"
900 forf=1tonf
910 print"[DBLU]";f;f1$(0,f
);":[GREEN]";:inputf1$(nr,f
)
920 nextf
930 print"[DOWN,RED]Any mor
e (Y/N)":getkeya$:ifa$<>"y"
thenreturn
940 goto 870
950 :
960 rem * insert a record *
```

```
970 print"[HOME2,CLR,RED,SP
C10,RVSON] INSERT A RECORD
"
980 gosub820
990 ifnr=maxthenprint"[BLAC
K,DOWN,SPC8,FLASHON]FILE FU
LL [FLASHOFF]- PRESS A KEY"
:getkeya$:return
1000 nr=nr+1
1010 input"[DOWN,BLUE]Enter
 record number :[ORNG]";rn
1020 ifrn<0orrn>nrthen1010
1030 gosub1110:print"[DOWN,
PURPLE]Record number";rn;"[
DOWN]"
1040 forf=1tonf
1050 print"[DBLU]";f;f1$(0,
f);":[GREEN]";:inputf1$(rn,
f)
1060 nextf
1070 print"[DOWN,RED]Any mo
re (Y/N)":getkeya$:ifa$<>"y
"thenreturn
1080 goto 990
1090 :
1100 rem * move records up
one place *
1110 forr=nr-1tornstep-1
1120 forf=1tonf
1130 f1$(r+1,f)=f1$(r,f)
1140 nextf,r
1150 return
1160 :
1170 rem * delete a record
*
1180 print"[HOME2,CLR,RED,S
PC10,RVSON] DELETE A RECORD
 "
1190 gosub820
1200 input"[DOWN,BLUE]Which
 record is to be deleted :[
ORNG]";dl
1210 if dl<1ord1>nrthen1200
1220 print"[DOWN,PURPLE]Rec
ord number";dl;"[DOWN]"
1230 forf=1tonf
1240 print"[DBLU]";f;f1$(0,
f);":[GREEN]";f1$(dl,f)
1250 nextf
1260 print"[DOWN,RED]Are yo
u sure (Y/N)":getkeya$:ifa$
<>"y"then1280
1270 gosub 1320:nr=nr-1
1280 print"[DOWN,RED]Any mo
re (Y/N)":getkeya$:ifa$="y"
then1180
1290 return
1300 :
1310 rem * move records dow
n *
1320 forr=dltonr-1
1330 forf=1tonf
1340 f1$(r,f)=f1$(r+1,f)
1350 nextf,r
1360 return
1370 :
1380 rem * edit a record *
1390 print"[HOME2,CLR,RED,S
PC10,RVSON] EDIT A RECORD "
1400 gosub820
1410 input"[DOWN,BLUE]Which
 record is to be edited :[O
RNG]";ed
1420 if ed<1ored>nrthen1410
1430 print"[DOWN,PURPLE]Rec
ord number";ed;"[DOWN]"
1440 forf=1tonf
1450 print"[DBLU]";f;f1$(0,
f);":[GREEN]  ";f1$(ed,f)
1460 print"[DOWN,DBLU]";f;f
1$(0,f);":[GREEN]";:inputf1
$(ed,f)
1470 nextf
1480 print"[DOWN,RED]Any mo
re (Y/N)":getkeya$:ifa$="y"
then1390
1490 return
1500 :
1510 rem * view records *
1520 print"[HOME2,CLR,RED,S
PC13,RVSON] VIEW RECORDS "
1530 gosub820
1540 re=1
1550 print"[HOME,DOWN,PURPL
E]Record number";re;"[DOWN]
"
1560 forf=1tonf
1570 print"[DBLU]";f;f1$(0,
f);":[GREEN]";f1$(re,f)
1580 nextf
1590 print"[HOME,DOWN21]";
1600 print"[ORNG]<-[RED].Ba
ck a record    [ORNG]->[RED
].Next record"
1610 print"[ORNG] G[RED].Go
 to a record    [ORNG] E[RED
].Exit to menu"
1620 getkeya$
1630 ifa$="e"thenreturn
1640 ifa$="[LEFT]"andre<>1t
henre=re-1:goto1550
1650 ifa$="[RIGHT]"andre<>n
rthenre=re+1:goto1550
1660 ifa$<>"g"then1620
1670 input"[DOWN,BLUE]Enter
 record :[ORNG]";re
1680 ifre<1orre>nrthen1670:
elsegoto1550
1690 :
1700 rem * sort file *
1710 print"[HOME2,CLR,RED,S
PC14,RVSON] SORT FILE "
1720 gosub820
1730 print"[PURPLE,DOWN]Fie
lds :"
1740 print:for f=1 to nf
1750 print"[DBLU]  f"[LEFT
].[GREEN]"f1$(0,f)
1760 next
1770 input"[DOWN,BLUE]Which
 field to sort by :[ORNG]";
sf
1780 if sf<1 or sf>nf then
goto 1770
1790 rem shell sort routine
1800 print"[DOWN,RED,RVSON]
WORKING[RVSOFF] :"
1810 lk=nr
1820 for z=0 to 1 step 0
1830 lk=int(lk/2)
1840 for lb=1 to lk
1850 ll=lb+lk
1860 for p=ll to nr step lk
1870 for f=1 to nf:d$(f)=f1
$(p,f):next f
1880 for q=p to ll step-lk
1890 for f=1 to nf
1900 f1$(q,f)=f1$(q-lk,f)
1910 next f
1920 if d$(sf)>f1$(q,sf)the
n for f=1 to nf:f1$(q,f)=d$
(f):next:q=ll
1930 next q
1940 if d$(sf)<=f1$(lb,sf)t
hen for f=1 to nf:f1$(lb,f)
=d$(f):next
1950 next p
1960 next lb
1970 if lk=1 then z=1
1980 next z
1990 return
2000 :
2010 rem * search for word
*
2020 print"[HOME2,CLR,RED,S
PC11,RVSON] SEARCH FOR WORD
 "
2030 gosub820
2040 print"[DOWN,BLUE]What
is the search word :[ORNG]"
;
2050 input sw$:in$=sw$:gosu
b2250:sw$=in$
2060 print:print"[RED,RVSON
,DOWN]SEARCHING[RVSOFF] :"
2070 print
2080 for r=1 to nr
2090 for f=1 to nf
2100 in$=f1$(r,f):gosub2250
2110 if instr(in$,sw$)=0 th
en goto 2190
2120 print"[DOWN,PURPLE] Re
cord number";r;"[DOWN]"
2130 print:for f1=1 to nf
2140 print"[DBLU]";f1;f1$(0
,f1);":[GREEN]";f1$(r,f1)
2150 next
2160 poke239,0:print:print"
[RED]Continue (Y/N)?"
2170 getkeya$:ifa$="n" then
 f=nf:r=nr
2180 f=nf:print"[BLACK]OK"
2190 next f,r
2200 print:print"[RED]Press
 Return";
2210 geta$:ifa$<>chr$(13)th
en2210
2220 return
2230 :
2240 rem * lower case conve
rtion *
2250 for c=1 to len(in$)
2260 a$=mid$(in$,c,1)
2270 if a$>="A" and a$<="Z"
 then mid$(in$,c,1)=chr$(as
c(a$)-128)
2280 next
2290 return
2300 :
2310 rem * add contents of
a field *
2320 print"[HOME2,CLR,RED,S
PC7,RVSON] ADD CONTENTS OF
A FIELD "
2330 gosub820
2340 print"[PURPLE,DOWN] Fi
elds :[DOWN]"
2350 print:forf=1tonf
2360 print"[DBLU]  ";f;"[LE
FT].[GREEN]";f1$(0,f)
2370 nextf
2380 input"[DOWN,BLUE]Which
 field to add up :[ORNG]";a
d
2390 ifad<1orad>nfthen2380
2400 print:tt=0:forr=1tonr
2410 tt=tt+val(f1$(r,ad))
2420 nextr
2430 print"[DOWN,DBLU]Total
 is[RED]";tt
2440 print"[DOWN,RED]Press
return"
2450 getkeya$:ifa$=chr$(13)
thenreturn:elsegoto2450
2460 :
2470 rem * save file *
2480 print"[HOME2,CLR,RED,S
PC13,RVSON] SAVE FILE "
2490 gosub820
2500 print"[DOWN,RED]Positi
on tape and press return"
2510 geta$:ifa$<>chr$(13)th
en2510
2520 open1,1,1,f1$
2530 print#1,f1$:print#1,nr
:print#1,nf
2540 forr=0tonr
2550 forf=1tonf
2560 iff1$(r,f)=""thenprint
#1,chr$(1):elseprint#1,f1$(
r,f)
2570 nextf,r
2580 close1
2590 return
2600 :
2610 rem * load a file *
2620 print"[HOME2,CLR,RED,S
PC13,RVSON] LOAD FILE "
2630 gosub820:clr:max=500
2640 input"[DOWN,BLUE]Enter
 filename :[ORNG]";f1$
2650 print"[DOWN,RED]Positi
on tape and press return"
2660 geta$:ifa$<>chr$(13)th
en2660
2670 open1,1,0,f1$
2680 input#1,f1$:input#1,nr
:input#1,nf
2690 dimf1$(max,nf)
2700 forr=0tonr
2710 forf=1tonf
2720 input#1,f1$(r,f)
2730 nextf,r:close1
2740 goto500
2750 :
2760 rem * print records *
2770 print"[HOME2,CLR,RED,S
PC10,RVS@N] PRINT RECORDS "
2780 gosub820
2790 print"[DOWN,RED] ARE Y
OU SURE (Y/N)"
2800 getkeya$:ifa$<>"y"then
return
2810 print"[CLR,DOWN3,DBLU,
SPC9]A.[RED]Print all recor
ds."
2820 print"[DBLU,SPC9]B.[RE
D]Print selected records."
2830 print"DOWN11,PURPLE] P
ress Selection:[DOWN,CT,UP]
<[LEFT,BLACK]";
2840 c$="ab":gosub760
2850 ifc=2then2990
2860 print"[CLR,DOWN3,RED,R
VSON,FLASH ON]PRINTING"
2870 open4,4:cmd4
2880 print:printchr$(17);ch
r$(14);"      Filename:";f1$
2890 print"[SPCS]";left$("[
CT24]",9+len(f1$));chr$(15)
2900 print:print
2910 forr=1tonr
2920 printchr$(17);"Record
number";r
2930 print:gosub3220
2940 print:print
2950 nextr
2960 print#4:close4
2970 return
2980 :
2990 open4,4:cmd4
3000 print:printchr$(14);ch
r$(17);"      Filename:";f1$
3010 print"[SPCS]";left$("[
CT19]",9+len(f1$));chr$(15)
3020 print#4
3030 print:input"[DOWN,BLUE
]Enter record to print :[OR
NG]";r
3040 ifr<1orr>nrthen3030
3050 print"[DOWN,PURPLE]Rec
ord number";r
3060 print
3070 forf=1tonf
3080 print"[DBLU]";f1$(0,f)
;":[GREEN]";f1$(r,f)
3090 nextf
3100 print"[DOWN2,RED]Are y
ou sure (Y/N)"
3110 getkeya$:ifa$<>"y"then
3170
3120 print"[BLACK]OK"
3130 cmd4
3140 print:printchr$(17);"R
ecord number";r
3150 print:gosub3220
3160 print#4
3170 print"[DOWN,RED]Any mo
re (Y/N)"
3180 getkeya$:ifa$<>"y"then
close4:return
3190 print"[CLR]":goto 3030
3200 :
3210 rem * print record *
3220 forf=1tonf
3230 printchr$(17);f1$(0,f)
;":";f1$(r,f)
3240 nextf
3250 return
3260 :
3270 end
```

Disk Sleeve Printer

for Plus/4, C16 and C64 with 1541 Disk Drive and Commodore compatible Printer

Knowing which files a particular disk contains, without having to load in the directory, can be a rather messy buisiness of squeezing the relevant information into the small space allowed on the lables supplied with the disk. An alternative method is to write the disk information on the corresponding disk's paper sleeve, in a similar manner as done with records and tapes. This simple BASIC program allows the directory of a 5.25" floppy disk to be listed to any Commodore compatible printer, in the format pattern of a disk sleeve.

The program may be run with either single or double sided disks, but the number of files contained on any side must not be more than 42 – once the front of the disk sleeve has been filled, the listing continues on the reverse side of the sleeve. Once the program has run, the result will be similar to figure 1. All that remains to be done is to cut around the dotted lines, fold and glue to form a sleeve.

Program notes

Type in the program as listed – the REM statements may be omitted if desired – and then save.

The program was developed for a Brother HR5-C 80 column dot matrix printer, however, it should run on other Commodore compatible dot matrix printer. The line spacing should be set to 1/6" if possible, although this is not essential.

Using the program

Connect the printer to your computer and load with paper – the pattern is printed in the centre of a piece of A4. Load in the SLEEVE PRINTER program, and RUN it. Select single or double sided disk when prompted, and insert the disk to be directoried into the drive (side A if you are using a double sided disk). After pressing any key, the program loads the disk directory and extracts the file names and file types, sorting them into a format ready for printing. If the double sided disk option is being used, then the disk should be reversed, when prompted, so that the operation can be repeated for the other side. If the single sided disk option is chosen, then the program will go direct to the print routine.

Before printing commences, one final prompt to load paper is given after which any key should be pressed to continue. Once printing has ended, remove the paper from the printer. Cut out the pattern along the dotted lines and then fold along the solid line. The flaps should then be folded over the back of the sleeve and glued – checking that the disk fits properly. After the glue has dried, insert the disk into its new home.

```
PROGRAM: DISK SLEEVE PRINTER


100 REM************************
110 REM*                       *
120 REM* DISK SLEEVE PRINTER   *
130 REM*                       *
140 REM*      BY J. HOYLE      *
150 REM*                       *
160 REM*          1987         *
170 REM*                       *
180 REM************************
190 REM*                       *
200 REM*    FOR USE WITH       *
210 REM*                       *
220 REM*   C64, PLUS4, C16     *
230 REM*                       *
240 REM*    AND COMMODORE      *
250 REM*                       *
260 REM* COMPATABLE PRINTERS    *
270 REM*                       *
280 REM************************
290 REM
300 REM
310 REM ******** MAIN LOOP *****
***
320 REM
330 DIM T$(50,1),A$(50,1)
340 S=0:REM SIDEA (S=0) SIDEB (S
=1)
350 PRINT "[CLR,DOWN,RIGHT]SINGL
E OR DOUBLE SIDED [S/D] "
360 GET S$
370 IF S$<>"S" AND S$<>"D" THEN
GOTO 360
380 IF S$="S" THEN GOTO 410
390 GOSUB 550:REM READ A SIDE
400 GOSUB 740:REM STORE A SIDE
410 GOSUB 550:REM READ B SIDE
420 GOSUB 740:REM STORE B SIDE
430 GOSUB 900:REM PRINT SLEEVE
440 PRINT "[CLR,DOWN,RIGHT]CUT O
UT AND GLUE PATTERN, AND THATS I
T!"
450 PRINT"[DOWN7,RIGHT]DO YOU WA
NT TO RUN AGAIN (Y/N) ?"
460 GET A$
470 IF A$<>"Y" AND A$<>"N" THEN
GOTO 460
480 IF A$="Y" THEN RUN
490 END
500 REM
510 REM ***** END OF PROGRAM ***
***
520 REM
530 REM ******** READ DISK *****
***
540 REM
550 PRINT "[CLR,DOWN8,RIGHT2]PLA
CE SIDE '"CHR$(65+S)"' OF THE DI
SK IN DRIVE."
560 PRINT TAB(11) "[DOWN4]THEN P
RESS ANY KEY."
570 GET K$
580 IF K$="" THEN GOTO 570
590 PRINT "[DOWN8,RIGHT5,FLASH O
N]PLEASE WAIT WHILST LOADING....
.[FLASH OFF]"
600 C=0
610 OPEN 15,8,15:OPEN 1,8,0,"$0:
*
620 FOR CO=1 TO 32
630 GET#1,I$
640 T$(C,S)=T$(C,S)+I$
650 NEXT CO
660 C=C+1
670 IF ST=0 THEN GOTO 620
680 CLOSE 1
690 CLOSE 15
700 RETURN
710 REM
720 REM ** GET FILE NAME AND STO
RE **
730 REM
740 PRINT "[CLR,DOWN8,RIGHT5,FLA
SH ON]PLEASE WAIT WHILST SORTING
.....FLASH OFF]"
750 A(S)=C
760 A$(0,S)=T$(0,S)
770 FOR CO=1 TO C-1
780 A$(CO,S)=" "
790 FOR L=1 TO LEN(T$(CO,S))
800 T$=MID$(T$(CO,S),L,1)
810 IF ASC(T$)<>34 THEN GOTO 840
920 A$(CO,S)=A$(CO,S)+MID$(T$(CO
,S),L,22)
830 L=LEN(T$(CO,S))
840 NEXT L:NEXT CO
850 S=1
860 RETURN
870 REM
880 REM ****** PRINT THE SLEEVE
*****
890 REM
900 PRINT "[CLR,DOWN3]"TAB(18)"D
.K...."
910 PRINT "[DOWN6,RIGHT3]POSITIO
N THE PAPER IN THE PRINTER"
920 PRINT "DOWN,RIGHT3]AND THEN
PRESS ANY KEY TO EXECUTE."
930 GET K$
940 IF K$="" THEN GOTO 930
950 OPEN 4,4:PRINT#4:P$=CHR$(18)
960 PRINT#4,P$"[CA]----[CR]---
--------------------------------
--------------------[CR]----[CS]
"
970 PRINT#4,P$"[S-,SPC3]* ****
********************************
****************** *[SPC3,S-]"
980 PRINT#4,P$"[S-,SPC3]* *";:
PRINT#4,P$"13"A$(0,0);:PRINT#4,P
$"38[RVSOFF]*";:PRINT#4,P$"40"A$
(0,1);
990 PRINT#4,P$"65[RVSOFF]* *[SPC
3,S-]"
1000 PRINT#4,P$"[S-,SPC3]* ***
********************************
****************** *[SPC3,S-]"
1010 PRINT#4,P$"[S-,SPC3]* *[S
PC53]* *[SPC3,S-]"
1020 FOR L=1TO18
1030 PRINT#4,P$"[S-,SPC3]* *";
:PRINT#4,P$"13"A$(L,0);:PRINT#4,
P$"40"A$(L,1);:PRINT#4,P$"65* *[
SPC3,S-]"
1040 NEXT L
1050 PRINT#4,P$"[S-,SPC3]* *[S
PC53]* *[SPC3,S-]"
1060 PRINT#4,P$"[S-]**** *****
********************************
****************** ****[S-]"
1070 PRINT#4,P$"[CZ]----[S+,S*
55,S+]----[CX]"
1080 PRINT#4,P$"[SPC5,S-]*****
********************************
******************[S-]"
1090 IF A(0)<19 AND A(1)<19 THEN
 GOTO 1190
1100 PRINT#4,P$"[SPC5,S-]*";:P
RINT#4,P$"13"A$(0,0);:PRINT#4,P$
"38[RVSOFF]*";:PRINT#4,P$"40"A$(
0,1);
1110 PRINT#4,P$"65[RVSOFF]*[S-]"
1120 PRINT#4,P$"[SPC5,S-]*****
********************************
******************[S-]"
1130 PRINT#4,P$"[SPC5,S-]*[SPC
53]*[S-]"
1140 FOR L=19 TO 42
1150 PRINT#4,P$"[SPC5,S-]*";:P
RINT#4,P$"13"A$(L,0);:PRINT#4,P$
"40"A$(L,1);:PRINT#4,P$"65*[S-]"
1160 NEXT L
1170 PRINT#4,P$"[SPC5,S-]*[SPC
53]*[S-]"
1180 GOTO 1220
1190 FOR L=1 TO 27
1200 PRINT#4,P$"[SPC5,S-]*[SPC
53]*[S-]"
1210 NEXT L
1220 PRINT#4,P$"[SPC5,[S-]****
********************************
******************[S-]"
1230 PRINT#4,P$"[SPC5,CZ]-----
--------------------------------
------------------[CX]"
1240 PRINT#4
1250 CLOSE 4
1260 RETURN
```

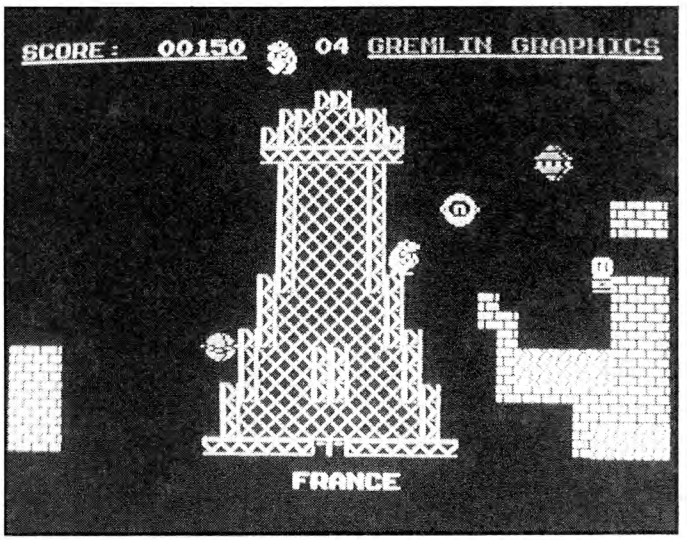
by J. Hoyle

Monty Mole is back again in this his final adventure. The hero of the C16's best platform games is in trouble again. Having escaped imprisonment for stealing coal to keep warm he has fled Britain and is lying low in Gibraltar. However, his whereabouts have been leaked to Intermole and the chase is on again.

Monty's only chance of freedom is to evade capture as he travels across Europe and collect enough money to buy the Greek island of Montos. Should he succeed in this almost impossible quest he will at last find sanctuary as nobody on the island knows him.

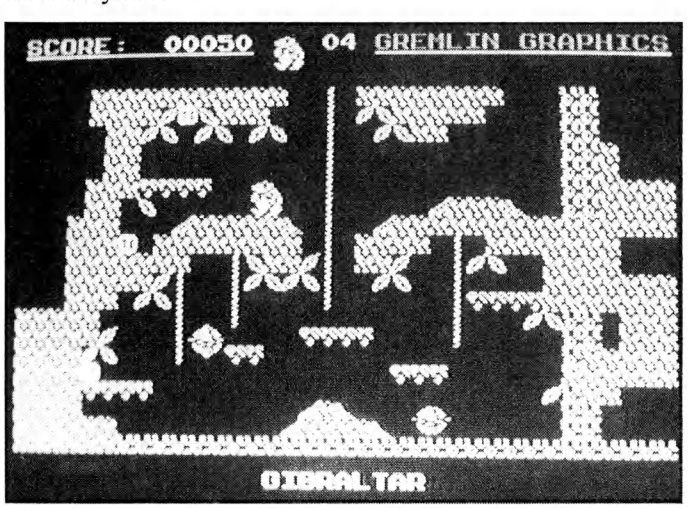
The cash in question is in the form of travellers cheques that appear on the screen as round discs marked TC. These are scattered about the screens that form Europe.

As in the other Monty Mole games these objects that you must collect are placed in the most obscure places that only someone as desperate as Monty Mole would try and get them.



Auf Wiedersehen Monty

The screen layouts are as fiendish and as difficult as those in Monty's previous adventures and consist of platforms to walk on, ropes to climb and hazards to avoid. These hazards take many forms ranging from less than subtle giant plungers that crush all in their path, to patches of rushing water that would drown our hero and critters that patrol. Any contact with these critters is fatal and will cost you one of your four lives. Cats have nine lives, moles have only four.



Each group of screens corresponds to a country in Europe and you'll soon find yourself climbing all over the famous landmarks. You begin the game standing on the Gibraltar rock as you make your escape into France across some mountains and up the Eiffel tower and so on into Spain, Italy, Germany, Austria, Switzerland and Greece until finally you reach Montos.

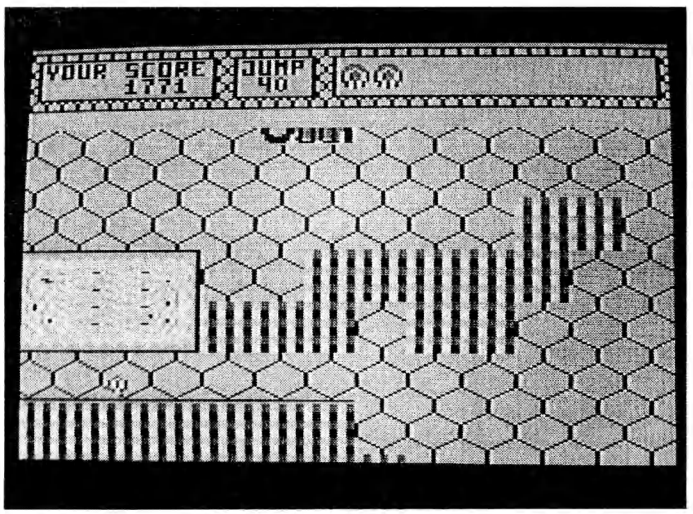
Unfortunately, you can only complete the game and buy Montos if you have collected all the travellers cheques so there's no missing out the difficult ones as you have to get them all.

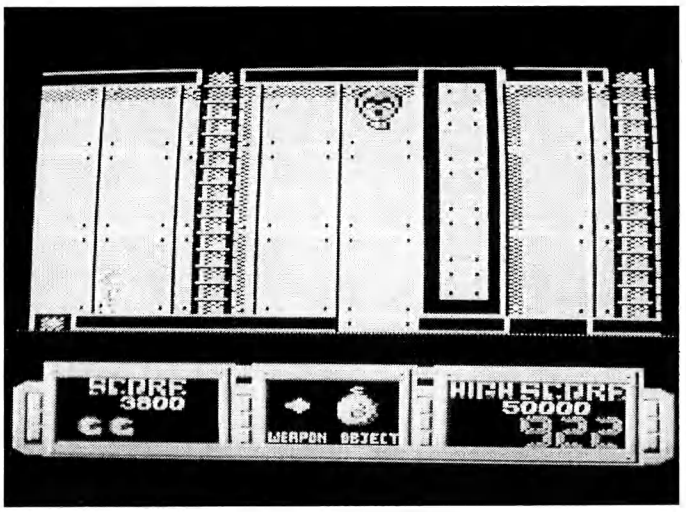
Some cheques are easier to get to than others as some are patrolled closely by critters are are just out of reach behind a plunger that you will have to watch for a few seconds to time your move to avoid being smashed.

Auf Wiedersehen Monty is a fitting end to the Monty trilogy and one of the most addictive and challenging platform games you can play.

Touchline:

Title: Auf Wiedersehen Monty. Supplier: Gremlin Graphics, Alpha House, 10 Carver Street, Sheffield, S11 4FS. TEL: 0742 753423.
Price: £7.99.





OMNIBUS

Gremlin's C16Plus4 Omnibus is an unbeatable compilation of ten games for the price of a single full priced game. Although the pack contains titles such as Planet Search (arcade action), Jetbrix (breakout game), Project Nova (space exploration) Tycoon Tex (six gun shoot-em-up), Rescue from Zylon (action) and Xargon Wars (alien attack) it is the other four games that make this a collection you must have.

Bounder is a superb arcade game where your timing and skill are put to the test as you bounce a ball along a course suspended in space. If you land on the right square you might collect some bonus jumps, extra power for your next leap and teleports however if you don't look before you leap you'll be swallowed up by giant mouth or activate a homing missile.

Future Knight is an unique style of platform game as it scrolls vertically. In the game you are Randolph the Future Knight and must rescue your beloved maiden from the evil clutches of Spegbott the Terrible. Your quest begins on board the SS Rustbucket which was the ship that was carrying Amelia that has now crashed on planet number 2749 in the Zragg system.

Twenty levels packed full of security droids will challenge you and your laser firing battle suit until you eventually find your way out to the planets surface and finally reach Spegbott's castle and a final battle with the giant killing machine the Henchman. On your way you will have to collect and use objects such as bombs that destroy the aliens on the screen and restores your constitution, confusers that stun the aliens for a few seconds, exit passes to reach the next level and a release spell to free Amelia and complete the game.

Trailblazer is a high speed race game over a course that hurtles out of the screen towards you. In some respects it is similar to Bounder as you must control a ball and land on squares that speed you up and give you extra bounce and avoid the gaps that will send you plummeting into oblivion as well as squares that slow you down and hurl you backwards.

In Trailblazer you barely have time to think leaving you strategy to your reactions you must complete each course within a time limit to survive to face the next one. There are 16 in all and each one is more challenging than the next. This is futuristic racing at it's best.

Finally, Footballer of the Year adds a new dimension to football games as you play a player instead of a manager. You begin as a 17 year old who has just joined a club in the lower divisions. Your aim is to score enough goals to be sold to better teams and from there become Footballer of the Year.

You start the game with 10 goal cards that you can play as your team plays it's matches. Each card is worth between one and three goals and each gives you a scoring chance that you must use your skill to convert. As your goals tally grows other teams will want you and you'll gradually move up the divisions until you reach the top, get a regular place in the England team and are voted Footballer of the Year.

An unbeatable compilation.

Touchline:
Title: Omnibus. Supplier: Gremlin Graphics, Alpha House, 10 Carver St., Sheffield S1 4FS.
Price: £7.99.

C16 & Plus/4 Software Offer

Do you wish that more C16 and Plus/4 programs were available on disk and cassette? Well here's a special offer of four packs of programs.

Due to length and complexity of the programs that are printed in this C16 and Plus/4 Special and regularly in *Your Commodore,* many people find that once they have typed them in they do not work. Usually, this is not the fault of the magazine, but rather, due to the program being typed incorrectly.

To help readers we do provide a Software for Sales service where the programs from several issues of Your Commodore are supplied on a single tape or disk. There have been three such compilations so far and we have added a fourth containing the programs from this supplement plus three from recent issues of *Your Commodore.*

C16 and Memory Expansion

The C16 and Plus/4 computers are almost identical, except for the fact that the C16 has far less memory than its big brother, the Plus/4. This compatibility means that programs for one of these computers will work on the other, as long as enough memory is available. The exception to this being programs that access the in-built software of the Plus/4, for example, the TRANSCRIPT program on the C16 C compilation.

How Much is the Software?

The price of the software is £5.00 for cassette and £7.00 for disk, this includes instructions. Orders should be sent to the address on the order form for Readers Services, they should NOT be sent to the editorial address.

Orders should be accompanied by a cheque or postal order for the correct amount made payable to Argus Specialist Publications.

We welcome orders from overseas readers. However, we do have to add a further £1.00 in order to cover the increased postal charges.

C16 Special A (7 programs)

The Monster Returns — an adventure set in the creepy world of Frankenstein's monster.

Keep it Simple — add icons, pull down menus and windows to your Plus/4.

Disk Monitor — talk directly to your disk drive.

Change Your Character — a C16 and Plus/4 editor to redesign your character sets.

C16 Assembler — out your C16 to serious use with this invaluable utility.

Break the Speed Limit — a high speed tape loader.

Plus/4 Dumper — obtain a hard copy of everything you do.

Tape Head Reader — examine the storage routine with this handy routine.

C16 Sound Sampler — sample a sound from your cassette and then edit it to produce amazing results.

C16 Special B (6 programs)

Dual Programming — work with two programs in memory at the same time.

Lower Case Graphics — improve the look of your programs by using the alternative character set.

Character Editor — devise your own character sets.

Cribbage — challenge your Plus/4 to a hand of this popular pub card game.

Spelling Checker — avoid those embarrassing mistakes with this ingenious program.

Word-pro Add-on — improve the Plus/4's built-in word processor.

C16 Special C (8 programs)

C16 Sprites — gives your C16 or Plus/4 sprites like those to be found on the C64. A demo routine is included to show you just what is possible.

+4 Animator — store a series of pictures in memory to create moving pictures.

Rebound — and excellent version of the latest breakout style games.

Disk menus — find and load your disk programs with ease.

Typro — turn your Plus/4 and printer into a powerful electronic typewriter.

Plus/4 Assembler — an excellent machine code assembler.

Transcript — owners of the Script Plus Cartridge can now convert their Plus 3 test files to work with this word processor.

Plus/4 extended basic — Add almost 40 new commands to the Basic on your Plus/4 or C16 with memory expansion.

C16 Special D (9 programs)

Money Plus/4 — organise your financial affairs with this superb program.

Plus/4 Database — a cassette based database for Plus/4 owners who don't want to buy a disk drive.

Fruit Machine — Holds, nudges and a time limit make this a challenging game.

Plus/4 Windows — add IBM PC Style windows to your programs.

Converter Plus/4 — converts your machine code to BASIC data statements.

Sleeve Printer — catalogue your disks by printing the directory on the disk sleeve.

Text80 — create 80 column displays on your Plus/4 screen.

ORDER FORM - PLEASE COMPLETE IN BLOCK CAPITALS

NAME	QTY	TAPE/DISK	ORDER CODE	PRICE
C16 Special A		Tape	YC16A	5.00
C16 Special A		Disk	YD16A	7.00
C16 Special B		Tape	YC16B	5.00
C16 Special B		Disk	YD16B	7.00
C16 Special C		Tape	YC16C	5.00
C16 Special C		Disk	YC16D	7.00
C16 Special D		Tape	YC16CD	5.00
C16 Special D		Disk	YC16DD	7.00
Overseas Post £1				
			Total	

NAME .....

ADDRESS .....

POSTCODE .....

In enclose a cheque/postal order for £ may payable to ARGUS SPECIALIST PUBLICATIONS LTD.

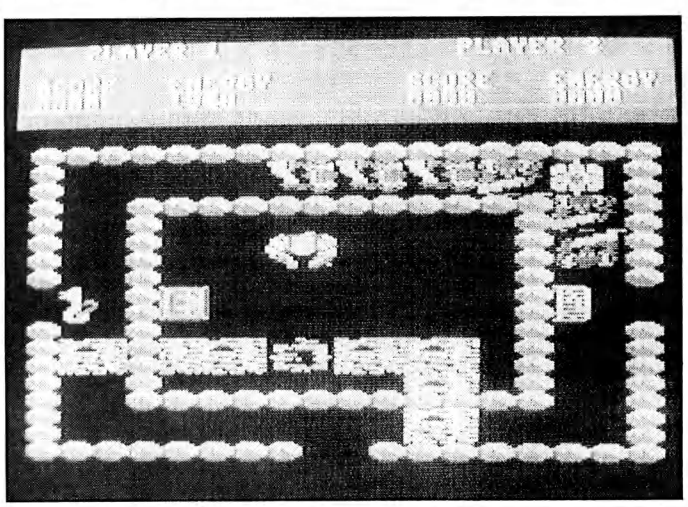
All orders should be sent to: YOUR COMMODORE, READERS SERVICES, ARGUS SPECIALIST PUBLICATIONS, 9 HALL ROAD, HEMEL HEMPSTEAD, HERTS HP2 7BH.
Please allow 28 days for delivery.

Storm is a real gem of a game and is the nearest thing that C16 and Plus/4 owners can get to the Gauntlet style of games. The game can be played by either one or two players who control Prince Storm and his friend the wizard Agravain Undead.

Our heros must run the gauntlet of the lair of the evil Una Cum's laboratory in search of Storm's wife Corrine who was kidnapped by the evil wizard. Una Cum has left his lair in search of a magic box called the Fear so now is your chance to rescue Corrine.

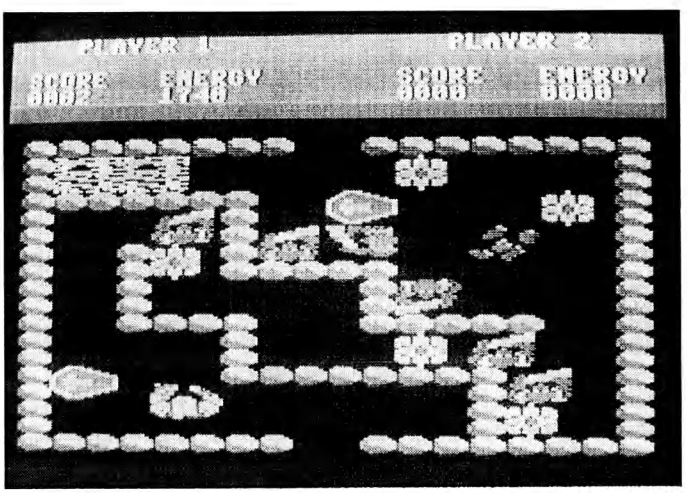
Una Cum's lair lies beneath the floorboards of the Abbey and so your view of the game is as you look down into his lair. As soon as you've read the instructions, the game loads in and your quest begins.

The screen display shows a top down view of the room you are in although some of it may be obscured by floorboards something you must watch or you may find yourself surrounded by Una Cum's minions that patrol the lair.



These minions are creatures that are spawned by generators which continue to produce monsters until they are destroyed. Therefore to clear a room you must kill all the monsters and destroy all the generators. However, when you leave a room the generators regenerate so be prepared for more combat if you go back on your tracks.

A room may have several generators in it so you may decide you may stand more chance of completing your quest by getting through a room as quickly as possible as every-time a monster touches you your energy is depleted. This begins at 2000 units but decreases for every second you are in the lair so you are going to have move quickly.



You can top up your energy levels by collecting food and bottles of restorative fluid but I wouldn't head for these if they're surrounded by generators as you're likely to lose more energy reaching them than you'll get for collecting them.

Magic Masks are a must as with these you can wield power magic to wipe out critters in your path. Scrolls and amulets have useful but weaker effects and destroy all monsters on a screen but these will be quickly replaced by the generators so you should use your time wisely.

You'll also find armour to protect you from the energy sapping touch of the monsters and three snake brooches that will unlock the door to the laboratory and lead to Corrine.

When you enter some rooms you may think that they are impossible as every exit and generator is enclosed by walls however you will also see a cabbala symbol in the floor and if you walk on this the walls will disappear revealing the exits and if you're not careful it will also swamp you monsters.

Storm is a challenging game particularly when played by two players working together.

Touchline:

Title: Storm. Supplier: Mastertronic, 2-4 Vernon Yard, Portobello Road, London, W11 2DX. TEL: 01-727 8070. Price: £1.99.

Text 80

Improve your C16/Plus 4's display with this handy utility by M.R. Everingham

Do you sit staring gloomily into your television screen, irritably playing with TEDMON, dreaming of CP/M and the IBM-PC? In short, does your Plus/4 lack character? If so, this could be the program for you. Before you get too excited, No this isn't a DIY Plus/4 — PC Upgrade, but it will increase the Plus/4's character — By 80 characters to be precise!

TEXT80 is a Machine-Code routine which will double the screen capacity of your Plus/4 Computer by enabling it to print 80 characters on each line, as opposed to the normal 40 characters. Another unique aspect of the program is that it works with the normal PRINT command in BASIC. This means that you can use all the text-formatting commands such as PRINT USING etc... Because the routine uses the High-Resolution Graphics Mode, it is not really suitable for use with the C16, as using this mode leaves you with only about 2K free for your programs. C16 users will either have to fork out forty quid for a 64K upgrade, or just be satisfied with dreaming!

Using the BASIC Loader Program

As TEXT80 is written entirely in Machine-Code, it needs to be POKEd into RAM in the normal way using a Loader Program.

The program is simple to use, just reset the computer, type it in, and type RUN! If there is an error in the data, the program will stop, telling you which line the mistake is in. When it finds that all the data is correct, the program will ask whether you wish to save the code to Tape or Disk, and on receiving your choice, will do so.

Please note that particular care should be taken in entering the POKE statements in lines 200 & 220.

By now you should have a working copy of (a) The BASIC Loader, (b) The Machine-Code Part1. (c) The Machine-Code Part2.

The TEXT80 Text Editor

Having successfully saved a copy of both parts of the machine-code, you should now enter the Text Editor program which demonstrates the capabilities of TEXT80. The program is entered in the normal way, and Tape-users must change ",8,1" in lines 10 and 20 to ",1,1".

On RUNning the program, the Tape or Disk should whirr into action, and the Machine-Code saved from the BASIC Loader will be loaded. Then after a brief pause, the screen should clear to white on black, and a menu appear at the bottom of the screen, with a copyright message at the top. On the second line of the screen there should be a solid cursor.

If all is well, you should be able to type from the keyboard and the text will appear on the screen 80 characters to a line! As well as just normal typing, the below features are also available:-

Switching TEXT80 On & Off

As shown previously, before any text is printed by TEXT80, the program must first be switched-on. This is done as follows:-

POKE 210,80 — Turns TEXT80 On.

When you do not wish text to be printed to the 80-Column screen, you must turn TEXT80 Off again. This is done thus:-

POKE 210,40 — Turns TEXT80 Off.

Positioning the 80-Column Cursor

As well as using the Cursor-Control Codes to move the cursor, you can POKE the X & Y values directly. This is done as follows:-

POKE 208,X — Set X-Coordinate to X.
POKE 209,Y — Set Y-Coordinate to Y.

Examples of printing in 80-Columns

```
10 GRAPHIC 1,1:POKE 208,80
20 PRINTCHR$(19)"HELLO THERE"
30 PRINT"28.5*3.14159=";28.5*3.14159
40 PRINTSPC(10)"TEN SPACES ACROSS"
50 PRINTTAB(10)"TEN TABS ACROSS"
60 PRINTUSING"$      ";13.25,32,100
70 PRINT"GOODBYE FOR NOW-..."
80 POKE 208,40
```

Using TEXT80 with Peripherals

If you have experimented enough, you will have noticed that when you print something to a printer or disk-drive using PRINT , with TEXT80 turned-on, the text goes to the 80-Column screen as well. This is not a bug, and provides a useful means of verifying what is being printed. As long as none of the printers internal control-codes are used, this feature can be used to preview printed-documents as they are sent to the printer!

Other commands and TEXT80

As we have seen, the main use of TEXT80 is with the BASIC PRINT command, but if TEXT80 is switched on, it can be used with any command that sends output via the CHROUT routine. These commands include DIRECTORY, MONITOR etc... The outcome of this is that you can have CP/M-style Disk Directories printed in 80-Columns, or 80 Character memory-dumps from TEDMON. Another interestin use is with the trace facility (Called by TRON). If TEXT80 is turned on, you can see what the program was doing on the 80-Character screen instead of getting the usual 40-Character Scrolling mess! All error-messages etc... will also appear on the 80-Column Screen.

Using TEXT80 from Machine-Code Programs

The technical details of the TEXT80 program and patch are shown below:-

Note that the Print-Screen option simple prints each line of text to a printer connected as device 4. It does not perform any formatting at all, so it should work with any 80-column printer (It was tested on a Citizen 120D-CBM).

Using TEXT80 from BASIC

I mentioned before that TEXT80 can be used easily from BASIC, and the Text Editor is written entirely in BASIC. Before TEXT80 can used several things must be done to patch it into the BASIC Operating System. The procedure for doing this is as follows:-

(1) POKE 55,197: POKE 56,249: CLR
(2) LOAD "80-Column M/C.1",D,1
(3) LOAD "80-Column M/C.2",D,1
(4) POKE 804,94:POKE 805,6
(5) POKE 210,80
(6) Do PRINTing
(7) POKE 210,40

OK, so what does all that do? The below should make things a little clearer.

(1) Reserves some space for the TEXT80 Program.
(2) Loads the TEXT80 Patch into RAM. (D) is the device.
(3) Loads the TEXT80 Program into RAM. (D) is the device.
(4) Patches the TEXT80 Program into the PRINT routine.
(5) Turns the 80-Column Mode on.
(6) Whatever you want!
(7) Switches back to 40-Column Mode.

Note that the value in (7) does not need to be 40, but can be any number apart from 80.

If the above procedure is to be performed from within a program, the following three lines must be at the very beginning of the program:

```
10 IF N=0 THEN POKE 55,197:-
POKE 56,249:CLR:N=1:LOAD "80-
COLUMN",D,1
20 IF N=1 THEN N=2:LOAD "80-
COLUMN M/C.2",D,1
30 POKE 804,94:POKE 805,6:POKE
210,40
```

Note that again, (D) is the Device number of the Tape or Disk.

When these program lines have been executed, the 80-Column mode is ready to use.

Programming using TEXT80

The way in which TEXT80 works is that whenever text needs to be printed, it first checks that the 80-Column X and Y Coordinates are valid, and if they are, prints that text to the 80-Column screen, also changing the Hires attributes to the current foreground and background colours. TEXT80 ALSO PRINTS TO THE 40-COLUMN SCREEN. The program prints to the 40-Column screen as well so that you do not get hopelessly lost when you forget to turn the 80-Column printing off when leaving a BASIC program. If printing goes off the screen the 80-Column screen will not be printed-to until X & Y are back within the valid ranges. Therefore before you experiment with 80-Column printing, you must home the cursor. The below program demonstrates this:

```
10 GRAPHIC 1,1
20 POKE 210,80
30 PRINTCHR$(19);
40 PRINT"Hello There!"
50 POKE 210,40
60 END
```

Note that the CHR$(19) in line 30 is the code for the HOME character, and could have been replaced by the Reverse-S character in quotes.

The CHROUT Patch

CHR$(19)(HOME)	—	Move cursor to (0,0).
CHR$(157)(LEFT)	—	Moves cursor left one character.
CHR$(29)(RIGHT)	—	Moves cursor right one character.
CHR$(145)(UP)	—	Moves cursor up one line.
CHR$(17)(DOWN)	—	Moves cursor down one line.
CHR$(13)(RETURN)	—	Moves cursor to left of next line.
CHR$(18)(RVSON)	—	Turns Reverse Printing on.
CHR$(146)(RVSOFF)	—	Turns Reverse Printing off.

You will probably have realised that the HOME character did more than its normal function — It also reset the 80-Column Coordinates. This is because TEXT80 decodes certain CHR$ Codes. These codes are as follows:

StartAddress: $065E
EndAddress: $0677
EntryAddress: $065E

The TEXT80 Printing Routine

StartAddress: $F9C6 (RAM Bank)
EndAddress: $FB8F (RAM Bank)
EntryAddress: $F9C6 (RAM Bank)

The TEXT80 Character-Set

To save on RAM, only screen-codes 0 to 45 are used by TEXT80. This gives all the standard characters in the Upper/Lower Character-Set. To save further on data, characters are stored two to one character, with four bits representing each character.

StartAddress: $FB90 (RAM Bank)
EndAddress: $FCFF (Ram Bank)
DataFormat: 8 Bytes x 4 Bits per char-acter.

Entry parameters for Printing Routine

Accumulator: Character-Code (PETSCII)
$00D0: X-Coordinate (0-79)
$00D1: Y-Coordinate (0-24)
Registers Affected:None
Other requirements: RAM-Bank in (interrupts Disabled)

(LEFT) — Move cursor left one character.
(RIGHT) — Move cursor right one character.
(UP) — Move cursor up one line.
(DOWN) — Move cursor down one line.
(RETURN)— Move cursor to left end of next line
(HOME) — Move cursor to the top-left corner of the screen.

The following key-presses have special functions:-

(DELETE) — Deletes last character typed and moves the rest of the current line back into the space created.
(INSERT) — Inserts a space into the current line, moving the line to the right and losing the last character on the line.
(CLEAR) — Asks you to confirm that you wish to clear the screen, and if you reply in the positive, clears the screen.
(CTRL-P) — Asks you to confirm that you wish to print the screen, and if you reply in the positive, does so.
(ESC) — Asks you to confirm that you wish to end the program, and if you reply in the positive, does so.

PROGRAM: 80 COL LOADER

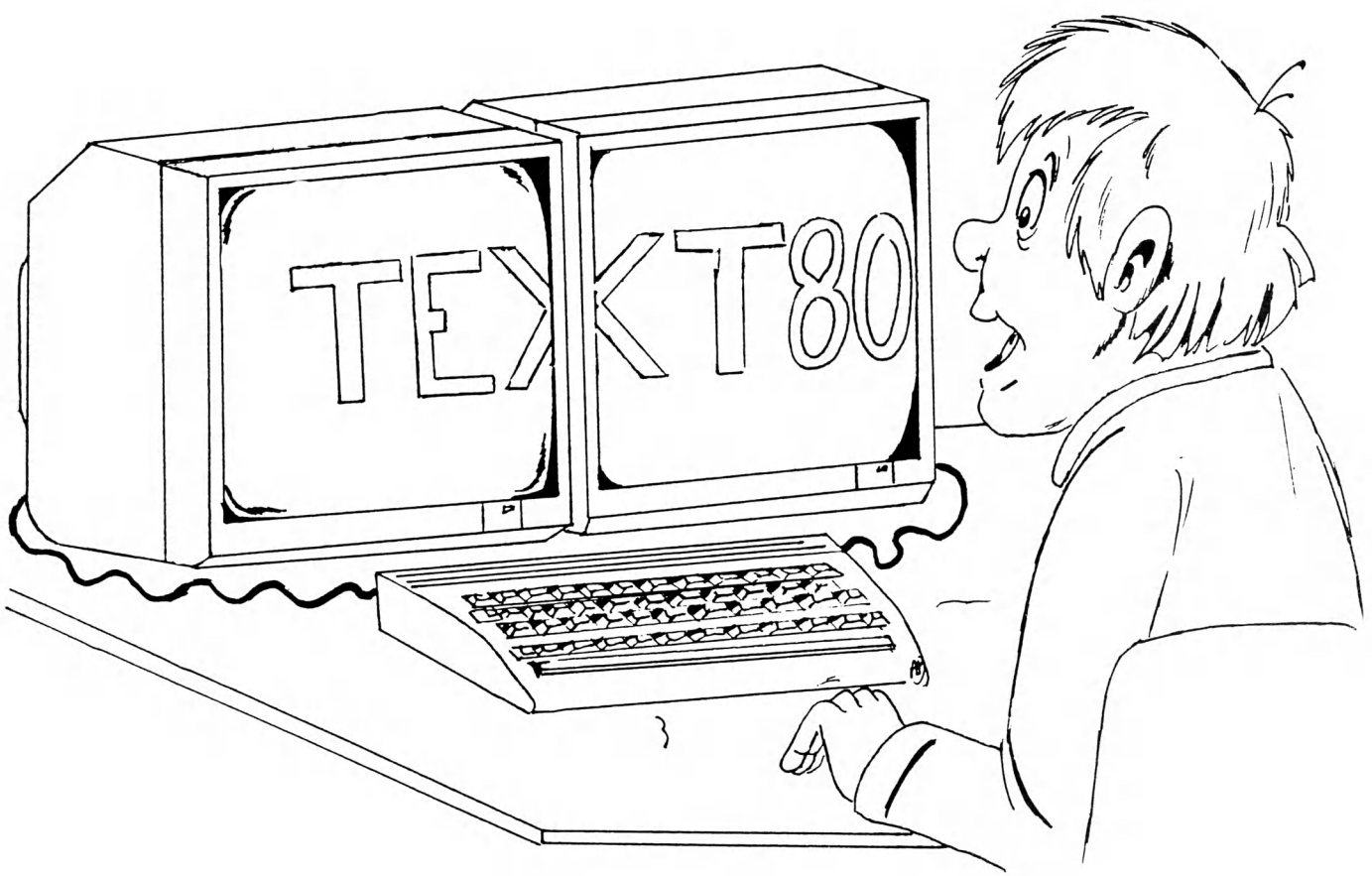
```
10 POKE 55,197:POKE 56,249:CLR
20 TRAP 160
30 PRINTCHR$(27)"R":RESTORE:A=16
30
40 FOR L=250 TO 280 STEP 10:C=0
50 PRINT"[HOME]STORING SECTION #
1 LINE"L
60 FOR Z=0 TO 7:READ D:POKE A+Z,
D:C=C+D:NEXT Z
70 READ V:IF C<>V THEN 150
80 A=A+8:NEXT L
90 A=63942
100 FOR L=290 TO 1320 STEP 10:C=
0
110 PRINT"[HOME,DOWN2]STORING SE
CTION #2 LINE"L
120 FOR Z=0 TO 7:READ D:POKE A+Z
,D:C=C+D:NEXT Z
130 READ V:IF C<>V THEN 150
140 A=A+8:NEXT L:GOTO 170
150 PRINT"[DOWN]DATA ERROR IN LI
NE"L:END
160 PRINT"[DOWN]"ERR$(ER)" ERROR
 IN LINE"EL:END
170 PRINT"[DOWN]DATA CORRECT - T
APE OR DISK? (T/D)"
180 DO:GET K$:LOOP UNTIL K$<>"":
IF K$="T"THEN POKE 208,1:ELSE PO
KE 208,8
190 PRINT"[DOWN]SAVING BASIC LOA
DER...":SAVE"80-COLUMN LOADER",P
EEK(208)
200 POKE 43,94:POKE 44,6:POKE 45
,120:POKE 46,6
210 PRINT"[DOWN]SAVING M/C PART
1...":SAVE"80-COLUMN M/C.1",PEEK
(208)
220 POKE 43,198:POKE 44,249:POKE
 45,0:POKE 46,253
230 PRINT"[DOWN]SAVING M/C PART
2...":SAVE"80-COLUMN M/C.2",PEEK
(208)
240 PRINT"[DOWN]SAVING COMPLETE
- RESET MACHINE.":END
250 DATA 72,165,210,201,80,240,4
,104,1076
260 DATA 76,75,236,120,141,63,25
5,104,1070
270 DATA 32,198,249,141,62,255,8
8,76,1101
280 DATA 75,236,0,0,0,0,0,0,311
290 DATA 8,133,220,138,72,152,72
,165,960
300 DATA 220,201,146,208,3,76,81
,251,1186
310 DATA 201,13,208,18,165,209,2
01,25,1040
320 DATA 144,3,76,81,251,169,0,1
33,857
330 DATA 208,230,209,76,81,251,2
01,17,1273
340 DATA 208,14,165,209,201,25,1
44,3,969
350 DATA 76,81,251,230,209,76,81
,251,1255
360 DATA 201,19,208,9,169,0,133,
208,947
370 DATA 133,209,76,81,251,201,2
9,208,1188
380 DATA 11,165,208,201,80,176,1
97,230,1268
390 DATA 208,76,81,251,201,145,2
08,12,1182
400 DATA 165,209,208,3,76,81,251
,198,1191
```

```
410 DATA 209,76,81,251,201,157,2
08,16,1199
420 DATA 165,208,208,7,169,79,13
3,208,1177
430 DATA 76,30,250,198,208,76,81
,251,1170
440 DATA 165,208,201,80,144,3,76
,81,958
450 DATA 251,165,209,201,25,144,
3,76,1074
460 DATA 81,251,165,220,201,32,1
76,3,1129
470 DATA 76,81,251,162,8,169,0,1
49,896
480 DATA 211,202,16,251,165,209,
133,213,1400
490 DATA 162,8,6,215,38,216,6,21
3,864
500 DATA 144,13,24,165,215,105,4
0,133,839
510 DATA 215,165,216,105,0,133,2
16,202,1252
520 DATA 208,232,165,208,74,144,
2,230,1263
530 DATA 211,133,213,6,213,38,21
4,6,1034
540 DATA 213,38,214,6,213,38,214
,165,1101
550 DATA 208,74,24,101,215,133,2
17,169,1141
560 DATA 0,101,216,133,218,24,16
5,218,1075
570 DATA 105,24,133,218,160,0,17
3,21,834
580 DATA 255,41,112,133,219,165,
134,41,1100
590 DATA 112,74,74,74,74,5,219,1
45,777
600 DATA 217,24,165,218,105,4,13
3,218,1084
610 DATA 173,21,255,41,15,133,21
9,165,1022
620 DATA 134,41,15,10,10,10,10,5
,235
630 DATA 219,145,217,6,215,38,21
6,6,1062
640 DATA 215,38,216,6,215,38,216
,24,968
650 DATA 165,213,101,215,133,217
,165,214,1423
660 DATA 101,216,24,105,32,133,2
18,169,998
670 DATA 0,133,214,165,220,201,1
28,144,1205
680 DATA 3,56,233,64,201,64,144,
3,768
690 DATA 56,233,64,133,213,70,21
3,144,1126
700 DATA 2,230,212,6,213,38,214,
6,921
710 DATA 213,38,214,6,213,38,214
,24,960
720 DATA 165,213,105,144,133,213
,165,214,1352
730 DATA 105,251,133,214,160,7,1
77,213,1260
740 DATA 166,212,208,39,41,240,3
2,98,1036
750 DATA 251,136,16,242,230,208,
165,208,1456
760 DATA 201,80,176,3,76,81,251,
169,1037
770 DATA 0,133,208,165,209,201,2
5,176,1117
780 DATA 2,230,209,104,168,104,1
70,40,1027
790 DATA 165,220,96,41,15,10,10,
10,567
800 DATA 10,76,52,251,166,211,24
0,22,1028
810 DATA 74,74,74,74,133,219,177
,217,1042
820 DATA 166,194,240,5,9,15,76,1
39,844
830 DATA 251,41,240,76,139,251,1
33,219,1350
840 DATA 177,217,166,194,240,5,9
,240,1248
850 DATA 76,139,251,41,15,69,219
,145,955
860 DATA 217,96,64,160,164,162,1
66,138,1167
870 DATA 102,0,128,128,198,168,1
68,168,1060
880 DATA 198,0,32,32,100,170,174
,168,874
890 DATA 102,0,0,0,70,170,138,19
8,678
900 DATA 130,140,128,132,192,172
,164,164,1222
910 DATA 174,0,40,8,42,44,42,42,
392
920 DATA 170,64,192,64,74,78,74,
74,790
930 DATA 234,0,0,0,196,170,170,1
70,940
940 DATA 164,0,0,0,198,170,170,1
98,900
950 DATA 130,130,0,0,102,136,132
,130,760
960 DATA 140,0,0,128,138,202,138
,170,916
970 DATA 76,0,0,0,170,170,170,17
4,760
980 DATA 74,0,0,0,170,170,74,166
,654
990 DATA 162,12,14,8,232,40,72,1
36,676
1000 DATA 238,0,110,130,130,194,
130,130,1062
1010 DATA 238,0,64,224,68,79,68,
64,805
1020 DATA 64,0,4,4,4,4,4,0,84
1030 DATA 4,0,160,170,174,10,14,
10,542
1040 DATA 0,0,74,98,132,68,36,20
0,608
1050 DATA 74,0,68,164,164,64,96,
128,758
1060 DATA 96,0,40,68,130,130,130
,68,662
1070 DATA 40,0,0,164,68,238,68,1
64,742
1080 DATA 0,0,0,0,0,14,0,32,46
1090 DATA 32,64,2,2,4,4,4,8,120
1100 DATA 72,0,68,172,164,164,16
4,164,968
1110 DATA 78,0,68,170,34,68,66,1
38,622
1120 DATA 228,0,174,168,172,226,
34,42,1044
1130 DATA 36,0,78,162,130,196,16
4,168,934
1140 DATA 72,0,68,170,170,70,162
,170,882
1150 DATA 68,0,0,0,64,2,64,2,200
1160 DATA 2,4,0,32,78,128,78,32,
354
1170 DATA 0,0,4,138,66,36,68,128
,440
1180 DATA 4,0,228,234,234,238,23
4,234,1406
1190 DATA 234,0,196,170,168,200,
168,170,1306
1200 DATA 196,0,206,168,168,172,
168,168,1246
1210 DATA 206,0,228,138,136,200,
138,138,1184
1220 DATA 134,0,174,164,164,228,
164,164,1192
1230 DATA 174,0,106,42,42,44,42,
170,620
1240 DATA 74,0,138,142,138,138,1
38,138,906
1250 DATA 234,0,164,170,234,234,
234,170,1440
1260 DATA 164,0,196,170,170,202,
138,138,1178
1270 DATA 134,0,196,170,168,196,
162,170,1196
1280 DATA 164,0,234,74,74,74,74,
74,768
1290 DATA 70,0,170,170,170,170,1
70,174,1094
1300 DATA 74,0,170,170,170,74,16
4,164,986
1310 DATA 164,0,224,32,32,64,128
,128,772
1320 DATA 224,0,0,16,0,0,0,16,25
6
```

PROGRAM: TEXT 80 EDITOR

```
10 if n=0 then poke 55,197:poke
56,249:clr:n=1:load"80-column m/
c.1",8,1
20 if n=1 then n=2:load"80-colum
n m/c.2",8,1
30 poke 804,94:poke 805,6:poke 2
10,40
40 color4,1:color0,1:color1,2:gr
aphic1,1:print"[SWLC][DISH]"chr$
(27)"c";:poke 210,80
50 dim l$(22):x=0:y=2:l$="
    "
60 for l=2 to 22:l$(l)=l$+l$+l$+
l$+l$+l$+l$+l$+"  ":next
70 print"[HOME]"tab(17)"80-Colum
n Text Editor By Mark Everingham
 1988."
80 draw 1,0,8 to 319,8:draw 1,0,
190 to 319,190
90 char 1,0,24,"
                              "
100 poke 208,8:poke 209,24
110 print"CLEAR - Clear Screen,
CTRL-P - Print Screen, ESC - End
 Program."
120 s$=mid$(l$(y),x+2,1):poke 21
0,40:printchr$(27)"O";:poke 210,
80
```

```
130 poke 208,x:poke 209,y:print"
[RVSON]"s$"[RVSOFF,LEFT]";
140 do:get k$:loop until k$<>""
150 if k$="[LEFT]"then 290
160 if k$="[RIGHT]"then 320
170 if k$="[UP]"then 350
180 if k$="[DOWN]"then 370
190 if k$=chr$(20)then 390
200 if k$="[INST]"then 430
210 if k$="[HOME]"then 470
220 if k$=chr$(13)then 480
230 if k$="[CLR]"then 500
240 if k$="[CTRL P]"then 540
250 if k$=chr$(27)then 650
260 printk$;:l$(y)=left$(l$(y),x
+1)+k$+mid$(l$(y),x+3)
270 x=x+1:if x>79 then x=0:y=y+1
:if y>22 then x=79:y=22
280 goto 120
290 if x=0 and y=2 then 140
300 prints$;:x=x-1:if x<0 then x
=79:y=y-1
310 goto 120
320 if x=79 and y=22 then 140
330 prints$;:x=x+1:if x>79 then
x=0:y=y+1
340 goto 120
350 if y=2 then 140
360 prints$;:y=y-1:goto 120
370 if y=22 then 140
380 prints$;:y=y+1:goto 120
390 if x=0 then 120
400 l$(y)=left$(l$(y),x)+mid$(l$
(y),x+2)+" "
410 poke 208,0:poke 209,y:printm
id$(l$(y),2,80);:x=x-1:if x<0 th
en x=79:y=y-1
420 goto 120
430 if x=79 then 120
440 l$(y)=left$(left$(l$(y),x+1)
+" "+mid$(l$(y),x+2,len(l$(y))-1
),81)+" "
450 poke 208,0:poke 209,y:printm
id$(l$(y),2,80);
460 goto 120
470 prints$;:x=0:y=2:goto 120
480 if y=22 then 120
490 prints$;:x=0:y=y+1:goto 120
500 prints$;:char 1,0,24,"[SPC40
]"
510 poke 208,30:poke 209,24:prin
t"Clear Screen? (Y/N)";
520 do:get k$:loop until k$="y"o
r k$="n"
530 if k$="n"then 90:else clr:go
to 40
540 prints$;:char 1,0,24,"[SPC40
"
550 poke 208,30:poke 209,24
560 print"Print Screen? (Y/N)";
570 do:get k$:loop until k$="y"o
r k$="n"
580 if k$="n"then 90
590 poke 210,40:open 4,4,7
600 for z=2 to 22
610 print#4,mid$(l$(z),2,80)
620 next
630 print#4:close 4
640 poke 210,80:goto 90
650 prints$;:char 1,0,24,"[SPC40
]"
660 poke 208,30:poke 209,24:prin
t"Abort Program? (Y/N)";
670 do:get k$:loop until k$="y"o
r k$="n"
680 if k$="n"then 90
690 poke 210,40:graphic0,1:print
"Program Aborted."
```



CONVERTER +4

By Jason Drew

Converter +4 is a BASIC program which converts sections of memory into BASIC DATA statements. This is very useful for people who submit their machine language programs to magazines. Instead of having a monotonous m/c listing where typing mistakes are easily made, you have BASIC DATA lines. There are eight hexadecimal numbers per DATA line, and a checksum. If the sum of the eight hex numbers doesn't agree with the checksum, the user will be told so, and which line contains the error. Then it's just a case of checking the line with the listing and correcting the mistake. Far better than searching through reams of m/c looking for one mistyped number.

When typing in 'CONVERTER +4' the line number used must be the same as the listing, otherwise the program will not work. The REMs from lines 0 to 9 can be omitted.

There are in fact two programs. The smaller of the two may not be needed, depending on where your m/c program resides. The main converter program resides from $1000 to $16DD. If your m/c program resides in this area, you will need to move the bottom of BASIC so the main converter program won't wipe out your m/c. That is what the smaller program is for – to move the bottom of BASIC. Upon running the program, you're asked to give the new location of BASIC. This should be above or below your m/c program enough to avoid it. After typing in the new location of BASIC, the program will change it accordingly and stop. Your m/c program should then be loaded. After that, load 'CONVERTER +4' and run it.

The main program will ask you the location of your m/c program. Then the screen goes blank while the m/c is being converted.

And that's it. You now have BASIC data lines with a checksum. Also, the program to poke the data is there, which will detect any typing errors. If no errors are found, you will be asked to give the filename of the POKEd m/c program. This must be at least two and at most sixteen characters long. The m/c program is then saved to disk or tape. To reload the program type: LOAD'(filename)',D,1 where D=8 for disk or 1 for tape. After the m/c is saved, the poke program and the data will remain. To save this, simply type in SAVE'(filename)', 8/1. 8 for disk and 1 for tape.

running of the program, but it does make it faster. By POKEing 65286 with 11, the screen is switched off and the program is processed faster as the computer doesn't have to worry about the screen display. By changing the value in line 10 of the variable OFF to 27, the screen is not turned off, and you will be able to see the program adding DATA lines to itself.

That's about it. This program should make life a little easier for anyone typing in yet another superb machine language program from Your Commodore.

The program

There are a number of important pokes used in this program. One is $EF (239) which is referred to as the 'keyboard queue index' in my memory maps. This, in effect, keeps count of the number of keys pressed before a GET command is encountered in a program.

Another few important locations are $0527 to $0530 (1319-1328). the 'keyboard queue' in computer talk. The keyboard queue index and the keyboard queue work hand-in-hand. 239 keeps track of how many keys have been pressed, and 1319-1328 keeps track of what those keys are. So if I want to add a line of converted data to the program, I clear the screen and print the line number and the data two lines from the top of the screen. The program then HOMEs to the top of the screen, pokes 239,2 and 1319-1320,13 (chr$ code for return) and then ENDs. When the program stops, the cursor is positioned on the line of data. Because of the previous pokes, a return is printed and the line of data is entered into the program. The cursor has then moved on to another line which says 'GOTO 20'. Another return is printed because of the pokes, and the program resumes, and so on until all the m/c is converted.

One more POKE of interest... 65286. This is not really vital to the

PROGRAM: CONVERTER +4

```
0 REM CONVERTER +4
1 REM   BY
2 REM JASON DREW
7 :
8 :
9 :
10 OFF=11:SC=27:GOSUB115:LN=400
15 POKE65286,OFF
20 SCNCLR:PRINT:PRINT:PRINTLN"DA
TA";
25 T=0:FORL=0TO7
30 P=PEEK(MC+L):T=T+P
35 PRINTRIGHT$(HEX$(P),2)",";
40 NEXTL
45 PRINTRIGHT$(HEX$(T),2)
50 IFMC>ETHENB5
55 LN=LN+1:MC=MC+8
60 PRINT"LN="LN":MC="MC":E="E
65 PRINT:PRINT:PRINT"PM="PM":GOT
O20"
70 POKE239,3:POKE1319,13:POKE132
0,13:POKE1321,13
75 PRINTCHR$(19);:END
80 END
85 SCNCLR:PRINT:PRINT
90 PRINTLN+1;"DATA END"
95 PRINT"LN="LN":MC="MC":E="E
100 PRINT:PRINT:PRINT"PM="PM":GO
TO185"
105 POKE239,3:POKE1319,13:POKE13
20,13:POKE1321,13
110 PRINTCHR$(19);:END
115 GRAPHIC0,1:COLOR4,2,1:COLORO
,2,3
120 COLOR1,3,7:PRINT"CONVERTER +
4":COLOR1,8,7:PRINT:PRINT
125 PRINT"ENTER HEX OR DEC START
 & END LOCATIONS?":PRINT"H/D"
130 GETK$:IFK$<>"H"ANDK$<>"D"THE
N130
135 PRINT:PRINT"ENTER START LOCA
TION.":PRINT">";:POKE19,64
140 INPUTS$:POKE19,0:PRINT:PRINT
```

```
145 PRINT"ENTER END LOCATION.":P
RINT">";:POKE19,64
150 INPUTE$:POKE19,0:PRINT
155 POKE65286,OFF:IFK$="D"THEN17
0
160 MC=DEC(S$)
165 E=DEC(E$):GOTO180
170 MC=VAL(S$)
175 E=VAL(E$)
180 PM=MC:RETURN
185 SCNCLR:PRINT:PRINT:POKE65286
,27
190 PRINT"DELETE-210":PRINT:PRIN
T
195 PRINT"RENUMBER11,1,0"
200 PRINT:PRINT:PRINT"10 MC="PM"
:PM="PM
205 POKE239,3:POKE1319,13:POKE13
20,13:POKE1321,13:PRINTCHR$(19);
210 END
215 RESTORE
220 T=0
225 FORL=0TO7:READH$:IFH$="END"T
HEN275
230 H=DEC(H$):T=T+H
235 POKEMC+L,H:NEXTL
240 T$=HEX$(T):READH$:IFRIGHT$(T
$,2)<>H$THEN250
245 MC=MC+8:GOTO220
250 SCNCLR
255 PRINT"ERROR IN DATA!"
260 DL=PEEK(64)*256+PEEK(63)
265 PRINT"ERROR IN LINE"DL:PRINT
270 END
275 SCNCLR
280 PRINT"* FINISHED *":PRINT
285 PRINT"ENTER PROGRAM FILENAME
.":PRINT">";:POKE19,64:INPUTPF$:
POKE19,0:PRINT
290 PF=LEN(PF$):IFPF<2ORPF>16THE
N275
295 PRINT"DISK OR TAPE?":PRINT"D
/T"
300 GETK$:IFK$<>"D"ANDK$<>"T"THE
N300
305 MC=MC+3:SCNCLR:PRINT:PRINT
310 HP=INT(PM/256):LP=PM-HP*256:
HM=INT(MC/256):LM=MC-HM*256
315 PRINT"POKE43,"LP":POKE44,"HP
":POKE45,"LM":POKE46,"HM:PRINT:P
RINT
320 IFK$="T"THEN330
325 PRINT"DSAVE"+CHR$(34)+PF$:GO
TO335
330 PRINT"SAVE"+CHR$(34)+PF$
335 PRINT:PRINT:PRINT
340 PRINT"POKE43,"PEEK(43)":POKE
44,"PEEK(44)":POKE45,"PEEK(45)":
POKE46,"PEEK(46)
345 POKE239,3:POKE1319,13:POKE13
20,13:POKE1321,13
350 PRINTCHR$(19);:END
```

PROGRAM: BASIC MOVER

```
0 REM CONVERTER +4 (BASIC MOVER)
1 REM    BY
2 REM JASON DREW
7 :
8 :
9 :
10 GRAPHIC0,1:COLOR4,2,1:COLOR0,
2,3
15 COLOR1,8,7:SCNCLR
20 PRINT"ENTER BASIC START LOCAT
ION":PRINT"IN HEX OR DEC?":PRINT
"H/D"
25 POKE239,0
30 GETK$:IFK$<>"H"ANDK$<>"D"THEN
30
35 PRINT:PRINT
40 INPUT"BASIC START LOCATION";B
S$
45 IFK$="H"THEN65
50 POKEVAL(BS$),0:BS$=STR$(VAL(B
S$)+1)
55 S=VAL(BS$):HS=INT(S/256):LS=S
-HS*256
60 GOTO75
65 POKEDEC(BS$),0:BS$=HEX$(DEC(B
S$)+1)
70 HS=DEC(LEFT$(BS$,2)):LS=DEC(R
IGHT$(BS$,2))
75 SCNCLR:PRINT:PRINT
80 PRINT"POKE43,"LS":POKE44,"HS
85 PRINT:PRINT:PRINT"NEW":POKE23
9,2
90 POKE1319,13:POKE1320,13:PRINT
CHR$(19);:END
```

GAME FINDER - See page 31 for more details

Mastertronic.
2-4 Vernon Yard, Portobello Road, London W11 2DX.
TEL: 01-727 8070.
Most games £1.99.
MAD games £2.99.
Best Buy — Storm.

Bandits at Zero Arcade ***
Dogfights over the sea as you take on an enemy carrier.

Battle Strategy **
A strategy war game between rival oil companies.

BMX Racers Sports ***
Complete five course to win the BMX Gold Cup.

Dingbat Arcade **
Hold off the hordes of aliens with your powerpack.

Fingers Malone Platform ***
15 levels of platform panic.

Frenesis Arcade **
40 levels of unrelented alien blasting.

Finders Keepers Platform ***
Avoid the ghouls and grab the treasure.

Formula 1 Simulator Sports ****
The best-selling racing car game.

GWNN Arcade **
Infiltrate alien bases to free scientists.

Kane Arcade ***
Wild west action.

Master Chess Strategy ****
A challenging chess program with plenty of options.

Oblido Arcade ***
A game where you will need your wits as well as your reactions.

On Cue Sports ****
Pool and snooker on the same cassette.

Powerball Arcade ***
Bounce yourself crazy with this addictive ball game.

P.O.D. Arcade ***
No frills, fast shoot-em-up.

Prospector Pete Arcade **
A digging game for gold but watch out for meanies.

Rockman Arcade ****
Boulder dash style of game.

Spectipede Arcade ***
C16 Centipede.

Speed King Sports *****
Bike racing at it's best.

Storm Arcade *****
Gauntlet gaming on the C16 / Plus 4

Tutti Frutti Arcade ***
An addictive version of the coin-op smash Mr Do!

Vegas Jackpot Strategy **
Nudge and shuffle in this fruit machine game.

Way of the Exploding Fist Arcade *****
The classic Kung-fu game.

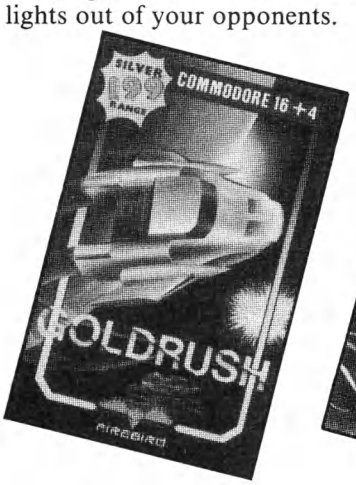
Terra Cognita Arcade ****
Over 100 screens in this challenging shoot -em-up.

Encore.
Elite Systems Ltd, Eastern Ave., Lichfield, West Midlands, WS13 6RX. TEL: 0543 414885.
All games are £1.99.
Best buy — Bomb Jack.

AirWolf Arcade ***
Take the controls of the gunship, Airwolf, to save scientists.

Bomb Jack Arcade *****
A superb coin-op conversion.

Frank Bruno's BoxingSports ****
Be the great man and bash the living day-lights out of your opponents.



Firebird.
64/76 New Oxford Street, London WC1A 1PS.
All Games are £1.99.
Besy Buy – Thrust.

Fury Arcade **
The classic game where you beat up aliens with a shovel.

Goldrush Arcade ***
Navigate the space corridors by blasting the asteroids.

Harvey Headbanger Arcade **
Battles in the cocktail bar.

Into the Deep Arcade **
Sideways scrolling shoot-em-up.

Netrun Arcade *
Man the twin gun emplacement to save the Empire.

Ninja Master Kung-fu *
Could you become a Ninja?

Shark Arcade **
Brave the deep armed only with a speargun in this underwater adventure.

Spikey Harold Arcade ****
Highly addictive arcade adventure.

Thrust Arcade *****
Skill and dexterity are needed in this classic.

Zolyx Arcade ****
Box off the screen and avoid the deadly touch of bouncing balls.

Gremlin Graphics.
Alpha House, 10, Carver Street, Sheffield, S1 4FS.
TEL: 0742 753423.
All Games £7.99.
Best Buy — Omnibus.

Auf Wiedersehen Monty Platform *****
The final adventures of Monty Mole.

Omnibus Compilation *****
10 games for the price of one including Trailblazer, Future Knight and Bounder.

Omnibus II Compilation ****
Monty on the Run and Kung Fu Kid star in this ten pack.

C16 Classics Compilation **
Dork's Dilemma, Tycoon Tex, Xargon Wars and Petals of Doom.

C16 Classics II Compilation **
Blagger, Monkey Magic, Timeslip and Xargon's Revenge.

C16 Classics III Compilation ***
Reach for the Sky, Sword of Destiny, Gullwing Falcon and Jetbrix.

Xcellor 8 Arcade **
Pilot your hover car on the trail of runners.

Trailblazer Arcade ****
High speed race game also part of Omnibus.

Gamefinder

While compiling this special C16 / Plus/4 supplement we were pleasantly surprised by the quantity and quality of the C16 and Plus/4 games we found. Most of which were available through the budget kings such as Mastertronic, Code Masters and Alternative software.

However, it has been difficult for C16 and Plus / 4 owners to track down these games and so we have compiled this gamefinder to help you. Under each company you'll not only find details of the address and phone number to contact but also a comprehensive list of their games including an overall rating and brief description. The ratings vary from one star (*) up to 5 stars (*****) for the pick of the games.

Armed with this gamefinder you should be able to find what you're looking for and answer critics who claim that there isn't any C16 or Plus/4 software.

Alternative Software
Units 3-6 Baileygate Industrial Estate, Pontefract, West Yorkshire, WF8 2LN.
TEL: 0977 797777.
All games are £1.99.
Best Buy — Arthur Noid.

Arthur Noid — Arcade — *****
Superb Arkaniod variant.

Fiends — Arcade — **
Rescue the scientists from the alien attack.

Liberator — Arcade — ***
Uridium style shoot-em-up.

Pheenix — Arcade — **
Battle with birds in this version of Phoenix.

Saboteur — Kung-fu — ****
Have you what it takes to survive this mammoth ninja arcade adventure?

Tower of Evil — Arcade — **
Top down adventure quest in the tower of evil.

Invasion Force — Compilation — **
Space Invaders, Tank Battle and Winnie the Witch.

Monkey Magic — Compilation — *
Monkey Magic, 3d Quasers and knockout (Breakout).

Galaxians — Compilation — ***
This coin-op classic along with Quick Draw and Mission Mars.

Space Freaks — Compilation — *
Space Freeks, Suicide Run and Meteorites.

Robin to the Rescue — Compilation — **
Versions of Hunchback, Pacman and Invaders.

Tazz — Compilation — **
Tazz is joined by Panzer duel and Trizons.

Code Masters.
Lower Farm House, Stoneythorpe, Southam, Warks., CV33 ODL.
All games are £1.99.
Best Buy — BMX Simulator.

BMX Simulator — Sports — *****
Seven courses to challenge BMX bikers

Danger Zone — Arcade — **
20 levels of aliens and asteroids.

G'man — Arcade — **
Jetpac powered scrolling action.